ISSN 0265-2919

80p

THE HOME COMPUTER COURSE 11

MASTERING YOUR HOME COMPUTER IN 24 WEEKS

An ORBIS Publication

IR £1 Aus $1.95 NZ $2.25 SA R1.95 Sing $4.50 USA & Can $1.95

CONTENTS

Next Week

- We look in depth at one of the first colour home computers — the Commodore Vic-20 — and how its special features all come from one chip
- Choosing your first computer, or upgrading your existing one, can be a difficult decision. We explain what to look for
- Arcade games are still the most popular application for home computers. We look at how they developed from the coin-operated machines

Editor Richard Pawson; **Consultant Editor** Gareth Jefferson; **Art Director** David Whelan; **Production Editor** Catherine Cardwell; **Staff Writer** Roger Ford; **Picture Editor** Claudia Zeff; **Designer** Hazel Bennington; **Art Assistants** Liz Dixon, Safu Maria Gilbert; **Sub Editor** Tracy Ebbetts; **Researcher** Melanie Davis; **Contributors** Tim Heath, Henry Budgett, Brian Morris, Mel Pullen; **Group Art Director** Perry Neville; **Managing Director** Stephen England; **Consultant** David Tebbutt; **Published by Orbis Publishing Ltd: Editorial Director** Brian Innes; **Project Development** Peter Brookesmith; **Executive Editor** Chris Cooper; **Production Co-ordinator** Ian Paton; **Circulation Director** David Breed; **Marketing Director** Michael Joyce; **Designed and produced by Bunch Partworks Ltd; Editorial Office** 39 Goodge Street, London W1; **© 1983 by Orbis Publishing Ltd: Typeset by Universe; Reproduction by Mullis Morgan Ltd; Printed in Great Britain by Artisan Press Ltd, Leicester**

HOME COMPUTER COURSE – Price UK 80p IR £1.00 AUS $1.95 NZ $2.25 SA R1.95 SINGAPORE $4.50 USA and CANADA $1.95

How to obtain your copies of HOME COMPUTER COURSE – Copies are obtainable by placing a regular order at your newsagent.

Back Numbers UK and Eire – Back numbers are obtainable from your newsagent or from HOME COMPUTER COURSE. Back numbers, Orbis Publishing Limited, 20/22 Bedfordbury, LONDON WC2N 4BT at cover price. AUSTRALIA: Back numbers are obtainable from HOME COMPUTER COURSE. Back numbers, Gordon & Gotch (Aus) Ltd, 114 William Street, PO Box 767G Melbourne, Vic 3001. SOUTH AFRICA, NEW ZEALAND, EUROPE & MALTA: Back numbers are available at cover price from your newsagent. In case of difficulty write to the address in your country given for binders. South African readers should add sales tax.

How to obtain binders for HOME COMPUTER COURSE – UK and Eire: Please send £3.95 per binder if you do not wish to take advantage of our special offer detailed in Issues 4, 5 and 6. EUROPE: Write with remittance of £5.00 per binder (incl. p&p) payable to Orbis Publishing Limited, 20/22 Bedfordbury, LONDON WC2N 4BT. MALTA: Binders are obtainable through your local newsagent price £3.95. In case of difficulty write to HOME COMPUTER COURSE BINDERS, Miller (Malta) Ltd, M.A. Vassalli Street, Valletta, Malta. AUSTRALIA: For details of how to obtain your binders see inserts in early issues or write to HOME COMPUTER COURSE BINDERS, First Post Pty Ltd, 23 Chandos Street, St. Leonards, NSW 2065. The binders supplied are those illustrated in the magazine. NEW ZEALAND: Binders are available through your local newsagent or from HOME COMPUTER COURSE BINDERS, Gordon & Gotch (NZ) Ltd, PO Box 1595. Wellington. SOUTH AFRICA: Binders are available through any branch of Central Newsagency. In case of difficulty write to HOME COMPUTER COURSE BINDERS, Intermag, PO Box 57394, Springfield 2137.

Note – Binders and back numbers are obtainable subject to availability of stocks. Whilst every attempt is made to keep the price of the issues and binders constant, the publishers reserve the right to increase the stated prices at any time when circumstances dictate. Binders depicted in this publication are those produced for the UK market only and may not necessarily be identical to binders produced for sale outside the UK. Binders and issues may be subject to import duty and/or local taxes, which are not included in the above prices unless stated.

Flights Of Fancy

Being able to simulate the complete functioning of a complex airliner or military plane means that pilots can receive most of their training on the ground

Cleared For Take-Off
'Bravo Alpha three two five, you are cleared for take-off on runway one two. Wind speed is one five knots, bearing one zero five. Call me when you are passing one thousand metres.'
But the runway in front of this Boeing 737 exists only as digital pulses inside Rediffusion's Novoview SP3 flight simulation computer system

As early as the 1940's, aeroplane pilots received some initial training not in real aircraft but in stylised mock-ups known as Link Trainers. Representing the first generation of flight simulators, these were fairly crude machines that set out only to give some idea of the effects of the aircraft's controls, and their purpose was to teach basic flying skills.

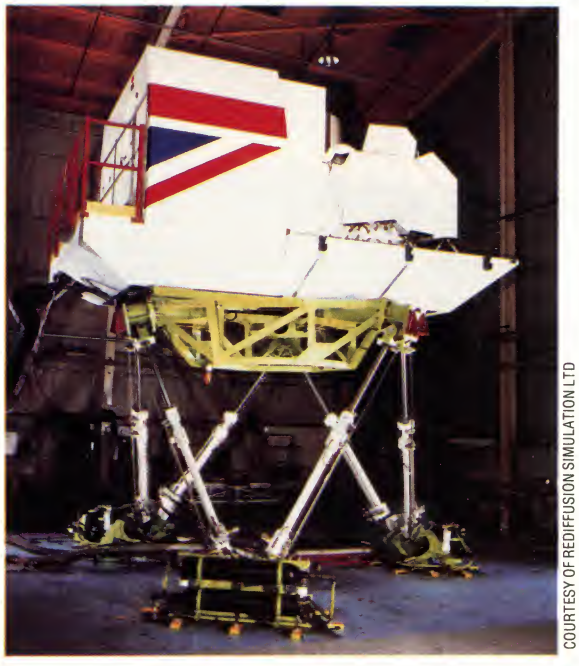
With the introduction of the first generation of multi-engined jet airliners such as de Havilland's Comet, it became apparent to airline operators that they would need a safer and more economical way of teaching their pilots to fly different aircraft types, other than actual in-flight training. Electronics manufacturers responded by producing the first computer-controlled simulators, using both analogue and digital techniques.

Mounted on hydraulic rams, these simulators replicated the cockpit sections of the craft that trainees were learning to fly. Computer control of the hydraulics systems allowed the mock-up to respond to movement of the flight controls just as the real aircraft would, and also allowed the instructor to set up 'emergency' situations.

The cockpit interior was an exact copy of the real thing, with complete instrumentation and a full set of controls. These controls were not only designed for the pilot, but for the entire flight-deck crew, since by this time it had become vitally important that the crew work as a team.

The only missing element was a simulation of the view that the pilot would have through the windscreen on take-off and landing. The first attempts at this centred on projected film, but the difficulties of constant re-orientation made this a far from perfect solution.

As so often seems to happen, the problem became really pressing at about the time that a solution was being proposed in another branch of technology. We have examined how Computer



COURTESY OF REDIFFUSION SIMULATION LTD

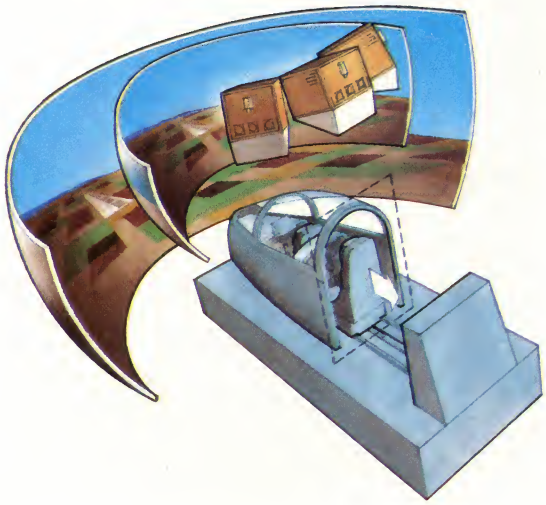
Planes On Legs
From the outside, flight simulators look like nothing more than rather odd-shaped boxes on angled hydraulic stilts. Once inside, however, you enter a completely different world — the flight deck of a modern aircraft

Generated Imaging (CGI) has revolutionised the production of animated films (see page 181). These same techniques are now used in flight simulators to provide pilots with moving images showing the appearance of an airport as the aircraft makes its approach. And because the animated image is held in software and can be loaded from disk, one simulator system can provide the pilot with a large repertoire of airports.

An even more impressive use of this technique is to be found in the flight simulators used by military pilots. Here the object is not to convert pilots from one aircraft type to another, for modern avionics systems are so refined that the aircraft virtually fly themselves. Pilot and observer training systems for military use are applied more to target recognition and weapons-aiming techniques, both air-to-ground and air-to-air. The basic need, however, is still to present the aircrew with a view from the cockpit that is as close to reality as possible.

At The Controls
Microsoft's Flight Simulator software puts the user at the controls of a Cessna 182 single-engined aircraft, flying from a representation of one of 20 actual airfields in the United States. A very comprehensive replica control panel occupies the lower half of the monitor screen, while the upper part represents a view of the 'outside world' given in full perspective. While the package includes a dogfight game, it is more of a true Computer Aided Learning program, aimed at teaching basic flying skills

When we talked about spreadsheet programs (see page 158), we described a technique called 'windowing' that allowed the operator to, as it were, pass the monitor screen over a larger sheet than it could contain. A similar technique is essential to the operation of an advanced flight simulator, in which the entire cockpit rig actually moves just as the real aircraft would in response to the controls and relative to the 'scene' outside. As the pilot goes into a diving turn, for example, so the appearance of the terrain in front of him must change.



Aerial Projections
Three three-gun (red, blue, green) TV projectors are used to throw the animated image of the outside world onto a screen in front of the pilot, by way of a collimating mirror, which adjusts for distortion caused by the curved back-projection screen

In an air-to-air combat simulation the situation is further complicated by the manoeuvres being performed by the adversary, and these must be randomised (or be taken from a large library of possible manoeuvres) so that the pilot is presented with a condition to which he responds not 'parrot-fashion' but with flexibility, making a fresh decision at every turn.

Military training schools go even further with their 'training by simulation'. They make use of extremely high-resolution images, satellite photographs from high-flying surveillance aircraft and the reports of agents, to produce an animated film showing detailed physical features of the terrain over which a pilot would fly on a low-level bombing mission.

Using this film in conjunction with a flight simulator allows 'whole mission' simulation to take place. This training method takes the pilot through the entire course of his sortie, from take-off to landing back at his home base, without his ever leaving the ground. The same flexibility of image generation that allows commercial pilots to 'practise' take-offs and landings at a variety of airports allows the military pilot to experience a wide variety of missions and targets.

Having been developed as a training aid for aircrew members, simulation techniques are now used by the Merchant Navy to familiarise watch-keeping officers with the appearance of channels and port approaches; by commercial divers, to help in object recognition in conditions of poor visibility; and by astronauts to simulate conditions in space. The power of Computer Generated Imaging techniques interacting with users under computer control makes it possible to replicate virtually any visual experience, which has made possible a whole new generation of games and entertainments.

Games programs that act as simple flight simulators are an obvious example of this phenomenon, challenging the user to make a successful flight, perhaps under a variety of adverse conditions: low fuel, poor visibility,

engine failure, and so on. But one should not think of these games as directly related to the professional systems that go by the same name, except perhaps in the case of Microsoft's Flight Simulator.

Any computer game, however — even the now obsolescent 'Pong', the precursor of them all — that represents a 'real-life' situation on the computer's television monitor can be regarded as a simulation of sorts. The only real difference lies in the complexity of the representation, and perhaps in the quality of the computer-generated image.

These simulations can come purely from the imagination, as Space Invaders and PacMan did. There the designer thought up the parameters of the game and translated them into possible courses of action. Or they can be more direct simulations of reality, such as the various car race games.

Not all home microcomputer-based flight simulation software is aimed at the games player. There are serious Computer Assisted Learning (CAL) packages available in the field of air navigation, air traffic control and flight planning, for example. Microsoft's Flight Simulator, however, available for IBM's Personal Computer, sets out to be both a recreational game and a serious exercise for would-be pilots.

The manual starts off by explaining that the package simulates all the instruments and equipment necessary under US Federal Aviation Authority rules for both day and night visual and instrument flying. The lower half of the screen represents the instrument panel of a Cessna 182 (the single-engined aircraft on which the simulation is based), while the upper part is

devoted to a medium-resolution animated colour picture of the view through the windscreen. This features pseudo-three-dimensional effects — that is, you get a reasonable sense of movement just from the changing perspectives.

Flight Simulator includes its own terrain map, which covers an area of approximately 2.5 million square kilometres (one million square miles) — basically the continental United States, with intrusions into Canada, Mexico and the Caribbean — encompassing some 20 different airfields in four main areas: New York/Boston, Chicago, Los Angeles and Seattle.

COURTESY OF REDIFFUSION SIMULATION LTD

'Enemy Aircraft . . . Bearing 030A . . .'
Computer Generated Images of this quality allow both air- and ground-crews to practise skills, like recognition in any sort of weather or lighting conditions, the instructor chooses. This example, produced on Rediffusion Simulations' Novoview system, shows an American A10 ground attack aircraft

It is theoretically possible to 'fly' right across the country, from New York to Los Angeles via Chicago, but as there are no refuelling points outside these populated areas, you would soon crash as a result of running out of petrol. Even if it were possible to 'carry' enough fuel, the simulator works in real time, which, with a maximum speed of around 240 kilometres per hour (150mph), means that the 'journey' would take around 20 hours — most of it in straight and level flight over open country. The package therefore permits you to 'jump' from one area to another.

Refinements include the ability to decide the time of day, the season (which affects the times of dawn and dusk), weather conditions including cloud cover, wind speed and wind direction, and even a factor for air turbulence.

Microsoft's Flight Simulator is based on the simplest type of aircraft. Nevertheless it is only through the use of highly efficient machine code that the program has been squeezed into 64 Kbytes. To simulate the entire flight deck of a modern wide-bodied jet, including actually moving the mock-up physically in response to movements of the controls, is a vastly more difficult task.

Classified Information

When designing any program that involves the storage or manipulation of information, it is important to pay close attention to structure, so that items of data needn't be duplicated

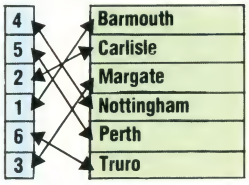
Shortest Route
A Midlands firm has interests in six towns, and has to send a lorry around them every month. The firm's computer can work out the most efficient route.

TownS(1)	"Barmouth"
TownS(2)	"Carlisle"
TownS(3)	"Margate"
TownS(4)	"Nottingham"
TownS(5)	"Perth"
TownS(6)	"Truro"

The names of the towns are in a sequential file on tape in alphabetic order. They are read from the file in this order into the array TownS( ). The computer works out that the best order is:

Nottingham	TownS(4)
Perth	TownS(5)
Carlisle	TownS(2)
Barmouth	TownS(1)
Truro	TownS(6)
Margate	TownS(3)

Rather than store the town names again in this order, using up more memory, the computer stores only the position numbers of the towns in the array TownS( ). Like this:



The list of numbers is an index to the array TownS( ). Other such indexes to the array could be created, each giving a different order to the names. The advantage of indexing like this is that the original file does not have to be sorted or duplicated: only the indexes need be sorted and stored

Anyone who has ever used a card index system (a library catalogue, for example) knows how useful it can be. If you have ever dropped the card box you know that the same cards out of order become maddeningly useless. A library contains a vast amount of information, but unless it's arranged in some sort of order its value as an information system is almost nil.

The essence of an information system is not the information itself but, rather, its organisation. Take, for example, this (fictional) item in a trade directory:

Smith, J., 15 High Street, Middletown.

On its own, its value as information is limited; if, however, you know that it comes from the Ironmongers' section of the directory, then it has a new significance that comes from the structure in which it is placed.

The simplest data structure is in the form of a file: a collection of data with common features. The name of the file reveals something about the information in it, and putting all that information together under that name makes it much easier to use. The file can be treated as one large unit of information, or as a particular grouping of smaller units. A book is a file; Mrs Beeton's autobiography is usually read as a whole, while her cookery book is usually read as a collection of individual recipes.

If the file is large, finding a particular piece of information may mean starting with the first item of the file, and scanning each item in turn until the right one is found. This is called sequential search or serial access, and a file arranged this way is a sequential file. A television programme is a sequential file of information, and so is a human conversation.

Sequential files are common because they are useful and cheap to implement, and because in many ways they mirror human methods of thought. However, they can become unwieldy and slow to use, so they are frequently divided internally into sub-files, which can be found directly without searching through the whole file. Books were once simple sequential files, but the invention of chapters, page numbers, and indexing transformed them. The chapters are sub-files of the book, and the pages are sub-files of both the chapters and the book.

A file that does not require sequential searching is called a direct access file. An album of songs on magnetic tape is a sequential file, while the same album on a long playing record is a direct access file: finding a song on the tape requires starting at the beginning and winding forward, while any track on the LP can be accessed directly by moving the pick-up arm across the LP to the start of the track.

Direct access depends on knowing where things are: in a book the index tells you what is where. Knowing exactly where things are means work (and, therefore, costs money). Indexing a book is an extra task for the author or publisher, and the information in the book may not warrant this expense; novels, for example, are not indexed, whereas textbooks usually are.

Computers process large quantities of information at speed, using a variety of data structures. Data has to be stored permanently on magnetic tape or disk in some structured way — typically in a sequential file — but in quite different ways in the computer's main memory.

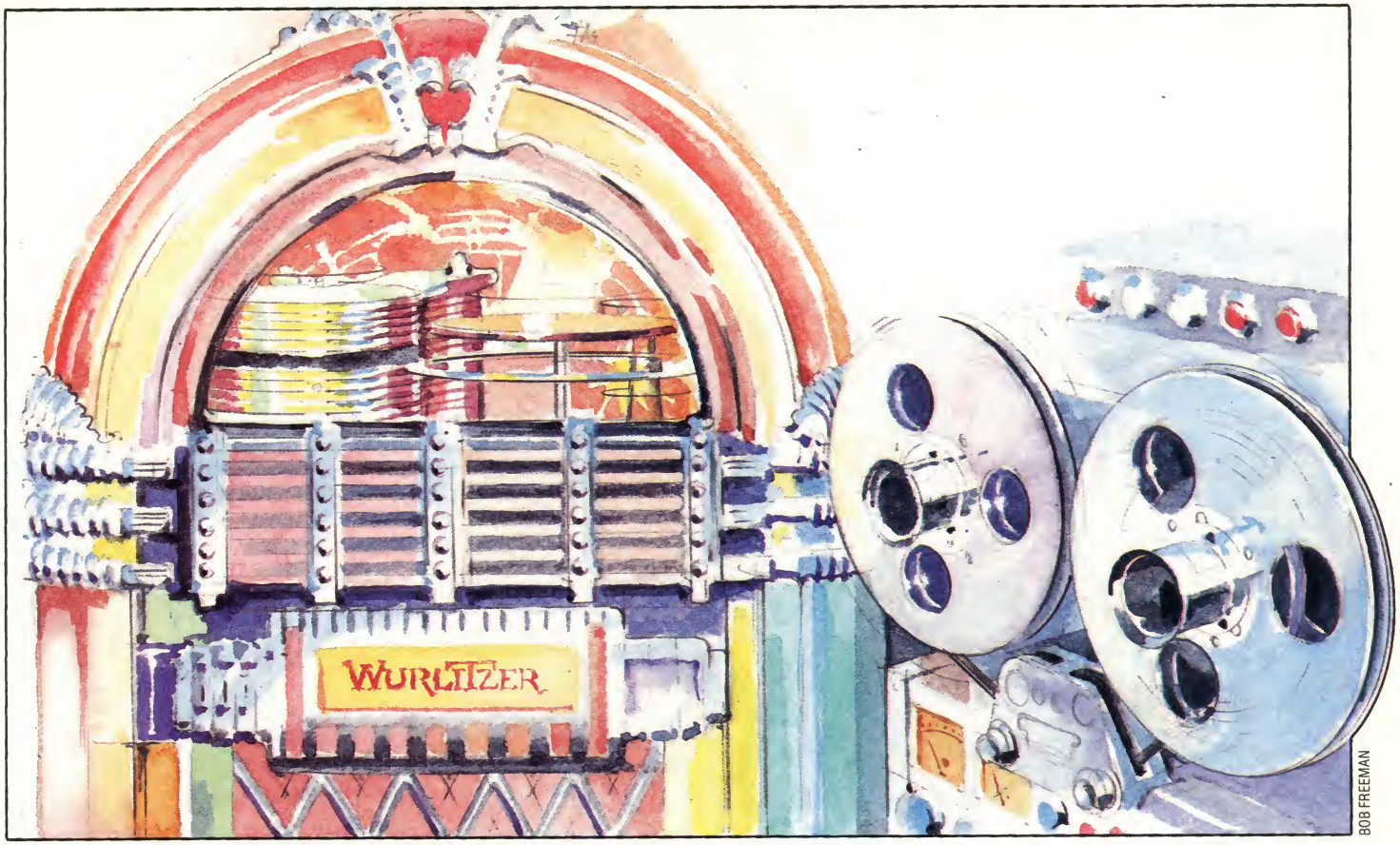
Suppose that a bakery has the addresses of all its shops in a sequential file on tape, and wants the computer to print delivery schedules for the drivers who take bread to the shops. The file on tape might look like this:

Atkinsons 22 High Street
Brown & Co 108 Alma Road
Edwards 49 Barking Lane
Wilson Bros 7 High Street
Wrights 65 Lower Road
Youngers 31 Parsons Hill

When the file is read from tape into main memory, each name and address will be stored in a numbered location of its own, and all these locations together form a block of memory with its own name, so the file in memory looks like this:

BLOCKNAME: Shops
1) Atkinsons 22 High Street
2) Brown & Co 108 Alma Road
3) Edwards 49 Barking Lane
4) Wilson Bros 7 High Street
5) Wrights 65 Lower Road
6) Youngers 31 Parsons Hill

Now the data items can be accessed individually by naming the block and the location within it. Shops(4), for example, contains Wilson Bros, 7 High Street. This structure is called an array (see page 194), the data structure most commonly used by computers for internal data processing. It is like a book with one piece of information on each page. Notice that this simple structure immediately



BOB FREEMAN

changes the way we look at data. A file of recognisable names and addresses has become a block of anonymous data. Computers do not need to know what a data item means, only where it is, and what to do with it.

The data in the array Shops( ) is in alphabetic order, but this is unlikely to be the most economical order in which to visit the shops. Suppose the computer works out that the best delivery schedule is:

1) Wilson Bros 7 High Street
2) Atkinsons 22 High Street
3) Edwards 49 Barking Lane
4) Brown & Co 108 Alma Road
5) Youngers 31 Parsons Hill
6) Wrights 65 Lower Road

This schedule might be stored in another array, but that would mean the same information is stored twice in the memory. Micro owners will know that RAM is limited, and it might be inconvenient or impossible to duplicate data in that way, so another method is needed.

If the actual data are replaced by their position numbers in the array Shops( ), then the delivery schedule looks like this:

BLOCKNAME: Deliveries
1) 4
2) 1
3) 3
4) 2
5) 6
6) 5

and what it really means is, 'First go to the shop whose details are stored in Shops(4), then go to Shops(1), then to Shops(3) . . .', and so on. The only significant information in the schedule is the order in which to visit the shops, so this is all that needs to be stored in the new array, Deliveries.

Deliveries( ) is now an index to the array Shops( ) for the purpose of deliveries. When printing this schedule the computer will use the numbers in the array Deliveries( ) to print the names and addresses from the array Shops( ) in the correct order.

In this simple exercise, information — the shops' names and addresses — has been manipulated but not changed by the different data structures imposed upon it. A data structure does not change the content of the data, but gives significance by associating it in an ordered way with other data.

Just as we can re-arrange the array Shops( ) by re-indexing it according to the array Deliveries( ), so can we construct other indexes to serve other purposes. When we discussed databases (see page 124), we noted that certain information could be selected by reference to pointers included in each individual record. In this way we can 'embed' in each record of the file Shops( ), a pointer that would indicate its place in the delivery schedule. We could further extend the record to include, for instance, a pointer into a file of standing orders — Atkinsons, for example, always have 48 white loaves, 12 wholemeal loaves, and so on. The production department could then run through the file extracting information relating only to the number of loaves that they need to bake.

On The Right Track

A juke box contains 200 songs on 100 record discs. It costs £2,000, and weighs 80kg (176lbs). To select any song, press three keys. Average time from SELECT to PLAY is 15 seconds. A reel of magnetic tape on a tape recorder may contain the same 200 songs. The tape recorder costs £500, and weighs 10kgs (22lbs). To select any song, rewind the tape, press PLAY, and wait. Average time from SELECT to PLAY is 1,500 seconds.

A juke box is a direct access device: it is fast, fairly specialised, and expensive. A tape recorder is a sequential access device: slow, much less specialised, but reasonably cheap. A cassette recorder connected to a microcomputer is a sequential access device, while a floppy disk drive is a direct access device, even though it may be used to store sequential files

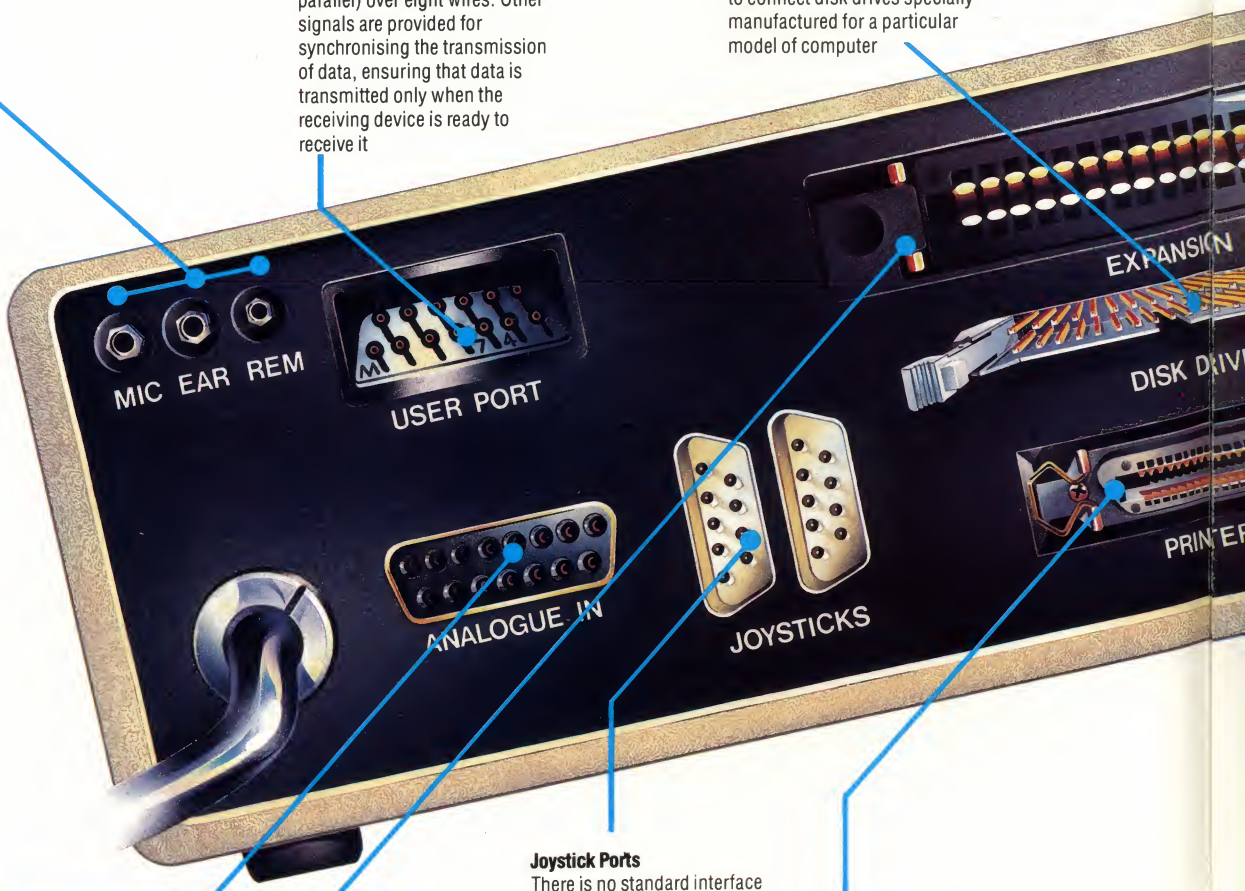
Influential Connections

You can't plug a square peg into a round hole, and you can't connect two computer devices together unless they have compatible interfaces

Cassette Interface
The cassette port or interface provided on most home computers is really a kind of serial interface. Because the data has to be recorded on ordinary audio cassette tape, using audio frequencies, high data-transfer rates are not possible. The interface circuitry takes bytes of data to be recorded from memory and converts each one into a stream of bits. 0 bits and 1 bits are rendered into two different audio tones. When tapes are loaded into memory, the tones are decoded by the interface circuitry and the resulting 1s and 0s assembled into eight-bit bytes for storage in memory. The cassette interfaces on most home micros are universal in the sense that any domestic tape recorder may be used successfully. The connector used is not standardised, but DIN sockets or mini-jacks are the most common

Parallel Port
This is a general-purpose parallel interface for connecting a micro to peripheral devices. All eight bits of each byte transmitted are sent together (in parallel) over eight wires. Other signals are provided for synchronising the transmission of data, ensuring that data is transmitted only when the receiving device is ready to receive it

Disk Drive Interface
Disk drives are usually connected to computers using a parallel interface. There is no standardisation and, as a general rule, it is only possible to connect disk drives specially manufactured for a particular model of computer



Analogue Input
Usually only found on the more expensive computers that are intended for educational use, an analogue input is useful for connecting the computer to devices in a laboratory, such as electrical temperature gauges or light-sensors. The interface will merely feature one or more lines that can accept and read a voltage in a specified range. It is up to the user to ensure that he doesn't connect the computer to a voltage outside this range, which could result in serious damage

Memory Expansion Port
This usually supports most if not all of the lines that come directly from the microprocessor — i.e. the address, data and control busses. This is where any additional memory will be plugged in and, on some computers, the manufacturer's peripherals as well. The connector is usually just a PCB edge-connector, though in some cases it may be a socket that can take an edge-connector, such as that on a games cartridge (which is really a form of ROM expansion)

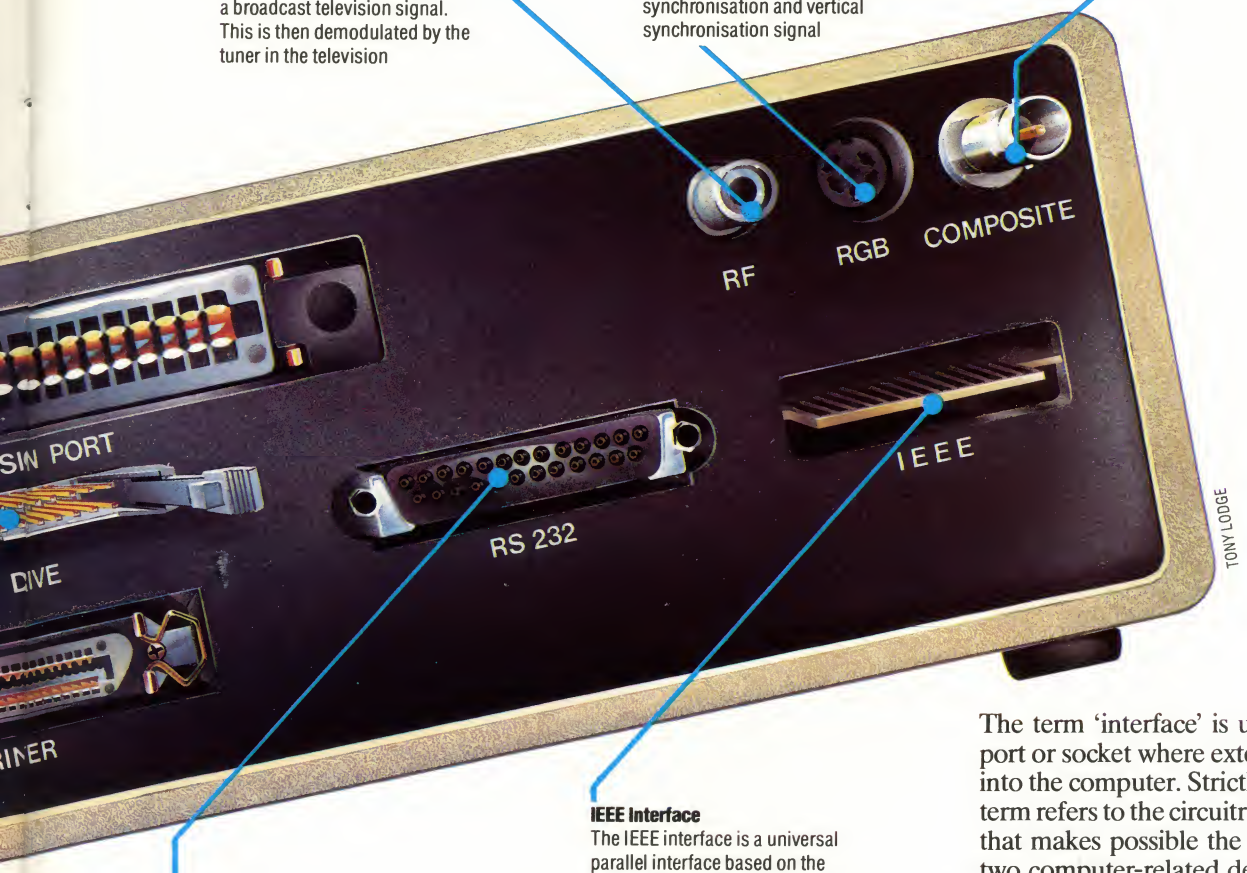
Joystick Ports
There is no standard interface for joysticks, though many manufacturers have adopted Atari's standard so that they don't have to manufacture their own joysticks. Most such interfaces simply have five active lines — one from each switch at the four extremities of the joystick's movement, and one for the fire button. Analogue joysticks, however, require a different interface that can accept a whole range of voltages to indicate the stick's exact position. Most computers feature more than one joystick port, though sometimes the same socket is shared by more than one device

Printer Interface
Printer interfaces are reasonably well standardised, following the system developed by the Centronics Corporation, so there is usually little difficulty in getting a printer with a Centronics interface to work with the printer interface on most computers. The signal levels, as well as signal functions, are also standardised at 0 and 5 volts for binary 0 and 1 respectively. The connectors used and the assignment of the signals to the various pins are not standardised, so you may have to wire up a special connector lead to connect a printer to the computer

Video Interface (RF)
All home computers are designed to be connected to a video display unit and for most this means a domestic black and white or colour television set. If the ordinary aerial input socket of the television is to be used, the video signal must first be RF (radio frequency) modulated so that it resembles a broadcast television signal. This is then demodulated by the tuner in the television

RGB Interface
Even better results are possible if the elements needed by the video monitor are kept separate. The RGB interface provides separate red, green and blue video signals, plus a horizontal synchronisation and vertical synchronisation signal

Video Interface (Composite)
Some televisions and many video monitors incorporate a composite or 'video' input that by-passes the RF demodulating stage, enabling better-quality pictures to be produced. All the elements of a standard video signal (chrominance, luminance, synchronisation signals, etc.) are present in the computer's output but this 'composite' signal does not need to be further modulated for processing by the television tuner



TONY LODGE

Serial Interface
Unlike many interfaces, the RS232 serial interface is theoretically precisely defined in a standard of the Electrical Industries Association. This standard specifies the type of connector to be used (a 25-pin 'miniature D' connector), the signals allocated to each pin, and the signal levels. Unfortunately, few manufacturers stick to the standard, and making serial devices work with a given computer can be difficult. Of the many pins present on the standard interface, usually only three are used: pin 2 to transmit serial data from the computer to the peripheral device; pin 3 to receive data transmitted from the peripheral device; and pin 7, the ground (earth) signal. Both transmitting and receiving devices need to be set so that the rate of data transmission and the format of the transmitted data are the same

IEEE Interface
The IEEE interface is a universal parallel interface based on the Hewlett-Packard Interface Bus, and now adopted as a standard by the Institute of Electrical and Electronics Engineers. The standard is well defined, both physically and electrically. Unlike other data interfaces (e.g., Centronics parallel and RS232 serial), which can connect only to one peripheral device at a time, the IEEE bus can connect up to 15 instruments (including the computer itself) simultaneously. Devices incorporating IEEE interfaces include printers, floppy disk drives, plotters, signal generators, voltmeters and other test equipment. Because it lends itself so well to use with test and measuring equipment, the IEEE bus is much favoured for use in laboratories and industrial establishments. At present only a few home computers offer an IEEE interface and some of these use a printed circuit board edge-connector instead of the standard IEEE connector

The term 'interface' is used loosely to mean the port or socket where external devices are plugged into the computer. Strictly speaking, however, the term refers to the circuitry and associated software that makes possible the connection between any two computer-related devices.

Internally, the computer communicates by sending data over 'busses' — sets of parallel conductors, each of which conducts a single binary signal. In most microcomputers there are three internal busses: an eight-bit data bus, a 16-bit address bus, and a control bus with, usually, signals consisting of between five and 12 bits that indicate the current condition of the CPU. Some of the control signals advise memory and peripheral devices whether the CPU wishes to retrieve data (read) or to deposit data (write). Others convey information from the outside into the CPU, informing it, for example, that a peripheral device has some data to input and requires attention.

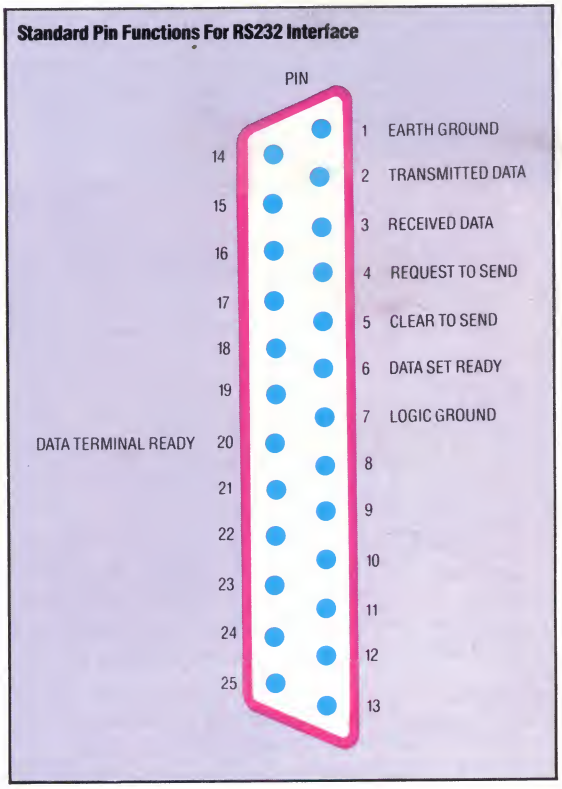
Internally, the computer generally handles information consisting of either eight bits or 16 bits at a time. Thus, if the CPU wants to retrieve the data in memory location 65535 (or FFFF, expressed in hexadecimal) it will set all 16 wires of the address bus to one to identify that location. If the contents of this memory location happen to be 182 (B6 in hexadecimal), this data will be placed on the data bus as the eight binary digits 10110110.

When data is transferred like this, eight or 16 bits at a time, the transfer is said to be 'parallel'.

Many peripheral devices are also designed to transmit or receive data in parallel. The interfaces provided for peripherals of this type are called 'parallel interfaces' and most computers provide at least one socket or connector specifically for 'parallel' devices.

Not all peripheral devices are able to receive or transmit data in parallel. Some use a single wire to communicate with the computer a bit at a time. Internally they still use the data as eight or 16-bit bytes, but the bits in each byte will be transmitted or received one at a time, starting from the 'least significant' bit in the byte and ending with the 'most significant' bit. Each byte is split up into a stream of bits, sent off one after the other, and re-assembled into a byte at the other end, using special parallel-to-serial and serial-to-parallel converter circuits.

Both serial and parallel interfaces can be made to convey information either out of the computer or into it. Computers usually include other interfaces, which either always send information out (for example, the television output circuit) or always accept incoming information (for example, joystick ports).



Serial Interface

There is a standard serial interface, known as the RS232 interface, for which every detail of the signal levels and pin assignments is defined. Even the type of connector is specified. Unfortunately, the standard is seldom fully adhered to and making serial connections work can be difficult. RS232 is a 'bi-directional' serial interface, with one pin (terminal) for transmitting data from the computer and one for receiving data.

The data is sent one bit at a time via the 'transmitted data' terminal and received via the 'received data' terminal. The stream of bits may take a number of standard formats but it doesn't matter which is used as long as both transmitting and receiving devices use the same one.

Since each byte of information is sent out as a serial stream of bits, the software controlling the interface must have a way of indicating when the first bit of the data starts and when the last bit has ended. The convention most commonly used has a single 'start bit' (0 in Boolean logic), followed by eight bits of data, followed by a single 'stop bit' (a logical 1).

The rate at which the data is transmitted needs to be specified in advance, otherwise the pattern of pulses representing the eight bits of data in 0s and 1s could be misinterpreted. This data transmission rate is known as the 'baud rate', named after the 19th-century French inventor Baudot, and pronounced 'bored'. Baud rates range between 75 baud and 9,600 baud. These figures correspond to 75 and 9,600 bits transmitted per second, and as there are usually 10 bits (including the start and stop bits) for each character, the character transmission rate is one tenth of the baud rate.

Parallel Interfaces

The parallel interface transmits or receives information one byte at a time, but in addition to eight 'data lines' it will also need to provide other signals so that the computer and the peripheral know when it is possible for data to be transmitted and when it is not. The commonest type of parallel interface is the 'Centronics' interface (named after an American printer manufacturer, Centronics Corporation), but this so-called standard is not rigidly adhered to. The type of connector used and the assignment of signals to particular pins differ widely from one manufacturer to another. Most Centronics interfaces provide at least the following signals:

DATA 0 to DATA 7	Eight wires to carry the eight bits of the byte being transmitted.
ADK	An input signal to the computer to indicate that the receiving device is ready to accept data.
GND	The 'ground' or earth lead that gives a common reference of 0 volts for both the computer and the peripheral device.
BUSY	A signal from the peripheral device to the computer indicating that the peripheral cannot accept data.
STROBE	An output signal from the computer that indicates to the peripheral that data is ready and should be read in.

Many other devices apart from printers adopt the quasi-standard Centronics parallel interface, and connection with the computer may involve no more than buying a special connecting lead or wiring up one of your own. As a general rule, no changes in the software needed to 'drive' the peripheral will be necessary.

Counting On Chance

Computers are completely rational, which makes them superior to humans for many tasks. Choosing a random number, however, is an irrational task, and therefore impossible for a computer

'Pick a number at random'. This is one of the easiest things for a person to do but, paradoxically, for a computer this simple task is impossible.

A random number is a number that is impossible under any circumstances to predict. A computer simply follows instructions and needs a reason for every action it takes, so any number it creates will be a consequence of a series of instructions. No matter how complicated these instructions are, the number is still theoretically predictable — to find it you just follow the instructions that the computer did — and so the number is not truly random.

Truly random numbers can, however, be obtained from physical processes. For example ERNIE (Electronic Random Number Indicator Equipment) uses the random motion of free electrons to pick the winners of premium bonds.

It was von Neumann who hit upon the idea of pseudo-random numbers — numbers that could be created mathematically. His method was to take a four-digit number, say 4321, and square it. From the resulting eight-digit answer, 18671041, he took the central four digits, 6710, as the random number. This then becomes the 'seed' for the next number to be created. So 6710 squared gives 45024100, the central digits of which are 0241, giving the next random number, and so on. This squaring process can continue indefinitely but since the total of possible different numbers with four digits is limited (9999) sooner or later the sequence must begin to repeat itself.

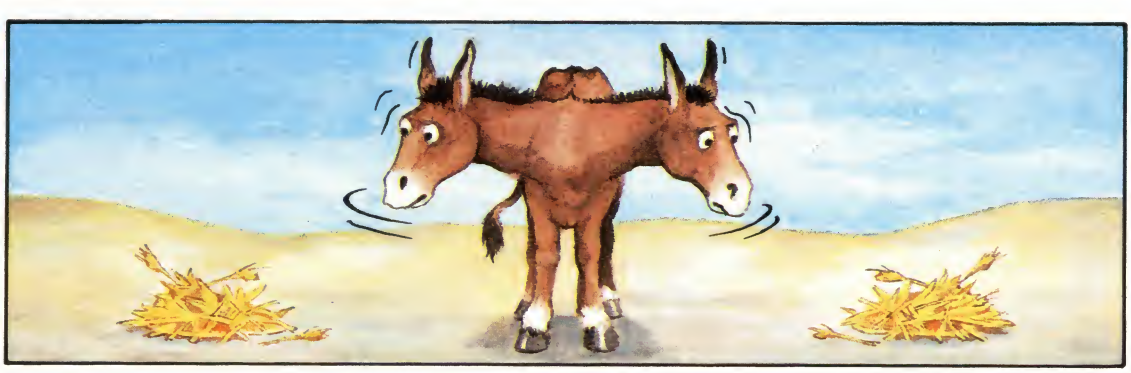
Modern computers use more involved methods that give better pseudo-random numbers. For example, the BBC Micro can generate a random number approximately every 1.5 milliseconds, and if the computer were generating these numbers continuously the sequence would take 150 days before it began to repeat itself.

Random numbers were first generated electronically for use by telephone engineers trying to simulate fluctuations in demand on exchanges. Today there are many uses, from computer games to simulation of randomly fluctuating processes, to evaluating difficult mathematical functions. For most purposes, modern algorithms for pseudo-random numbers can be considered truly random.

KEVIN JONES

The Monte Carlo Method

This method was invented by John von Neumann and uses random numbers to calculate the answers to mathematical problems. In mathematics you often need to calculate the area bounded by a curve and this in a way is analogous to finding the area of an island when you have only a map of the coastline. The map of Britain is shown in a box of a known size that is to be bombarded by random points. Since the positions of the points are randomly generated they will fall over the total boxed area and the number that falls within the landmass is proportional to the land area of Britain. Using 40 random points we found 24 fell at sea and 16 on the land. Hence the area of the land is: $16/40 \times (1050 \times 550) = 16/40 \times 577{,}500 = 231{,}000$ sq km. Using a greater number of random points would result in a closer approximation to the true figure of 229,523 square kilometres



DAVID HIGHAM

The Rational Ass

Imagine you put a totally rational donkey midway between two identical piles of hay. The donkey cannot choose the larger pile because the two piles are identical, and it cannot choose the nearer one because it is exactly midway between the two. So perhaps it goes to the one it glances at first. But why does it glance at one rather than the other? Since there is no reason to choose one pile rather than the other, the donkey just stands there and starves to death. Similarly, computers cannot generate truly random numbers, because they do everything strictly according to reason

Sinclair ZX81

In 1981 this tiny computer was considered revolutionary, both in terms of design and price. Since then its price has fallen further, but not its stature

The Sinclair ZX81 is the cheapest computer with a screen display and is an upgrade of the ZX80. The number of chips is reduced by the use of one ULA (Uncommitted Logic Array) chip to handle functions that previously required many components (see box). It is the Model T Ford of home computers: black, with no knobs, buttons or switches. The external power supply unit plugs into the side of the wedge-shaped case. Other connections are for cassette, television lead and peripherals. The Sinclair ZX Printer is also the cheapest means of getting 'hard copy' or listings, though the silver electro-sensitive paper that it uses is relatively expensive. The ZX81's screen, unlike other monochrome displays, prints black characters on a white background, and in upper case only. The Sinclair BASIC differs from the more standard Microsoft version, but is well suited to beginners.

The ZX81's keyboard is a striking feature. It is merely a printed picture of a keyboard with pressure-sensitive pads underneath. As such it is not suitable for touch typing, though it redeems iself to some extent when entering programs. When the computer is first switched on, the cursor is displayed as a K. This means that a 'keyword' such as PRINT or LET will be printed in full on the screen when the P or L buttons are pressed. These keywords are inscribed on the top of each key. Similarly, the cursor can be put into 'function mode' (displaying an F). When in this mode, pressing a key will print the function keyword printed above or below that key. For entering normal text, there is the L or 'letters' mode.

Though small and cheap, the ZX81 can still justifiably claim to be a computer and it has been around long enough for a vast collection of software to grow up around it. Enthusiasts have seen the tiny one Kbyte of standard memory as a challenge and have even managed to squeeze in programs like Adventures and chess. An extra 16 Kbytes of memory can be purchased in the form of a little plastic box that plugs into the back of the computer. However, this device has caused considerable aggravation for some users because the connector is not very secure and the slightest movement can result in all your data being lost. Inventive ZX81 users discovered that double-sided sticky tape helped, though Sinclair Research have now withdrawn the old RAMpack and replaced it with a more reliable one.

The ZX81 has also spawned hundreds of small companies offering all manner of extras. Some produce bigger and more reliable RAMpacks, connectors for non-Sinclair printers, and colour displays. There are even replacement cases that feature better keyboards and room to take most of the extras or add-ons inside them. Conversion is a simple matter of removing the PCB from the old casing, and placing it in the new one.

All in all the machine is the cheapest introduction to computing available and is also an ideal machine to buy if you're not sure you want to invest £200 or more in a home computer. Children love the keyboard. It is good enough to have found a place in nearly every primary school in Britain and will continue to be the cheapest computer for quite some time.

ZX81 Keyboard
In order to produce a microcomputer with an integral keyboard for an economy price, Sinclair adopted a 'sealed multiple membrane', which has been criticised ever since for its insensitivity and lack of 'feel'. The need to pack the keyboard into a small space dictated that optimising techniques should be used, which resulted in each individual key having a variety of effects. The 'L' key, for example, will return that single character, the assignment 'LET', the function 'USR', or the symbol '=', depending on which mode is in use at the time

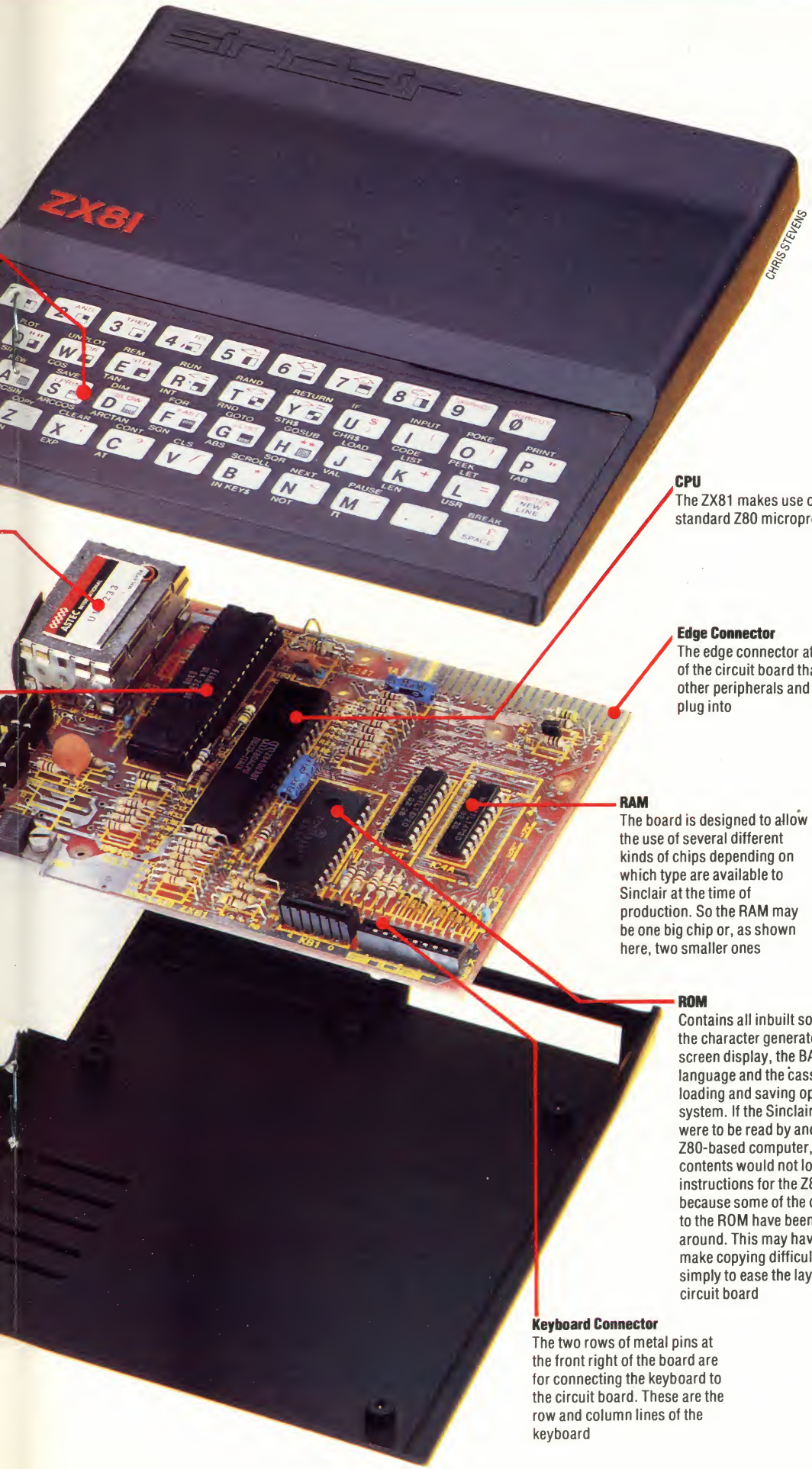
Uncommitted Logic Array (ULA)
Sinclair call this the 'Sinclair Computer Logic' chip. It is a special chip that contains the equivalent of several smaller 'logic' chips. These have a jargon name, 'glue' chips, because they 'glue' the important parts of a computer together. This ULA is the difference between the ZX80 and the ZX81. In the ZX80 there are about 12 small 'glue' chips, which meant the ZX80 cost more to build and has more likelihood of failure. ULAs, however, are difficult to design, and like ROMs, the design must be final and complete before any can be manufactured

RF Modulator
For standard TV sets. Provides video output

Connectors
Three miniature jack connectors for power, cassette input and cassette output. Some programs exist that produce music from the cassette output socket

Quantity Versus Quality

When Sir Clive Sinclair decided to launch a low-cost microcomputer for the home market, he was forced to make certain compromises. Chief amongst those was perhaps the keyboard, which has long been considered the weakest point of both the ZX80 and 81. As a result, one suspects, many users came to see the machine as ideal for running 'bought-in' software, rather than entering lengthy programs of their own, and a great deal was quickly available, especially for the games market. Hard on the heels of the inevitable versions of Breakout, Space Invaders and caterpillars that eat each other, came machine code de-bugging aids and, surprisingly, a variety of business-orientated packages, notably spreadsheets, though these need the additional RAMpack. BASIC is not the only language available for the ZX81. In addition, Sinclair offer FORTH (see page 150), and an Assembler for the Z80 microprocessor



CPU
The ZX81 makes use of a standard Z80 microprocessor

Edge Connector
The edge connector at the back of the circuit board that all the other peripherals and memory plug into

RAM
The board is designed to allow the use of several different kinds of chips depending on which type are available to Sinclair at the time of production. So the RAM may be one big chip or, as shown here, two smaller ones

ROM
Contains all inbuilt software: the character generator for the screen display, the BASIC language and the cassette loading and saving operating system. If the Sinclair ROM were to be read by another Z80-based computer, its contents would not look like instructions for the Z80. This is because some of the data lines to the ROM have been shuffled around. This may have been to make copying difficult, or simply to ease the layout of the circuit board

Keyboard Connector
The two rows of metal pins at the front right of the board are for connecting the keyboard to the circuit board. These are the row and column lines of the keyboard

ZX81

PRICE
£39.95 (special offer with the 16K RAMpack — £45)

SIZE
175 × 168 × 43mm

WEIGHT
300g

CLOCK SPEED
1MHz

MEMORY
8K ROM containing the BASIC interpreter and operating system. 1K RAM of which 123 bytes is taken up for the system variables, the rest being shared by your BASIC program and the screen display. The RAM can be expanded by 16K by inserting the Sinclair RAMpack

VIDEO DISPLAY
Black and white, 32 × 24 rows of text, 64 × 44 graphics. The bottom two rows are not for use as a normal display but show what is being typed into the computer

INTERFACES
Cassette I/O, power input and TV output are standard connectors. The edge connector at the back accepts other peripherals and extra memory

LANGUAGES SUPPLIED
Sinclair BASIC

OTHER LANGUAGES AVAILABLE
FORTH, Assembler for Z80 machine code

COMES WITH
Power supply adaptor, TV lead, cassette lead, documentation

KEYBOARD
40 pressure-sensitive pads, each able to produce one of several things: a BASIC command word, an alphanumeric character or a graphic symbol

DOCUMENTATION
The single manual is a very clear and concise introduction to using this computer. It assumes no knowledge whatever about computers, has lots of examples, exercises and summaries at the end of each chapter, but is devoid of interesting programs. The chapters from 23 onwards describe the computer in technical detail and provide hints on using machine code and a full list of all the system variables used by the operating system

Keeping Control

All versions of Basic feature 'control structures' that govern the flow of a program. Some machines, however, offer a wide range of alternatives, with subtle differences

The first 10 parts of the Basic Programming course have covered almost all of the more important aspects of the BASIC language. In this issue we will present an overview of the topics we have covered so far, deal with a few interesting asides and give some pointers to where we shall go next.

First the overview: a high-level language such as BASIC provides the user with a set of instructions that are translated internally into a form the computer can understand. Any computer program can be written using just two simple patterns, called 'constructs'. These are 'sequence' constructs and 'control structures' of which only two are essential in BASIC: IF . . . THEN . . . ELSE and WHILE . . . DO. Most other computer languages provide considerably more.

The sequence construct allows the task to be broken down into a set of sub-tasks that perform the main task when executed in sequence. The size of the sub-tasks depends on the language; in BASIC the sub-tasks are represented by the statements written on each line, and the sequence is represented by the line numbers. Thus, if the task is to multiply the value assigned to a variable by 10, the sequence we could use might be:

```
110 INPUT N
120 LET N = N * 10
130 PRINT N
```

In addition to sequence constructs, we also need control structures. These are constructs that alter the order of execution of statements in a program.

The simplest control structure provided by BASIC is GOTO. This is an unconditional jump (or branch) that re-directs the execution of the program to a specified line number without a test or condition having to be satisfied. GOSUB is also an unconditional branch, but the program will always RETURN to the point immediately after the GOSUB and its use in structured programming is perfectly acceptable.

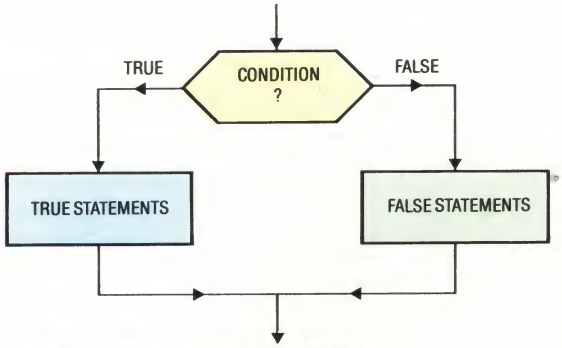
The IF . . . THEN . . . ELSE control structure is available in BASIC. It takes the form of the IF . . . THEN statement and has the following syntax ('syntax' is the computer jargon for 'form'):

IF (specified condition) is true THEN execute specified statement (ELSE) execute the next statement

Note that in standard BASIC, the ELSE part of IF . . . THEN . . . ELSE is implied. In some BASIC dialects and in certain other languages, PASCAL for example, ELSE forms part of the statement.

IF . . . THEN . . . ELSE (IF . . . THEN in BASIC) performs one of two sub-tasks depending on whether a certain condition is true or not. Consider the following program, which is designed to find the square roots of numbers input from the keyboard unless a 'flag' value of –9999 is input (in order to terminate the program):

```
10 PRINT "INPUT A NUMBER"
20 INPUT N
30 IF N = –9999 THEN GOTO 70
40 LET S = SQR(N)
50 PRINT "THE SQUARE ROOT OF ";N;" IS ";S
60 GOTO 10
70 END
```



The IF . . . THEN . . . ELSE Control Structure
If the condition is True, the True statements will be executed. If the condition is False, the False statements will be executed

What line 30 is really saying here is 'IF it is true that N = –9999 THEN go to the end of the program ELSE (if it is not true that N = –9999) execute the next line of the program to find the square root'.

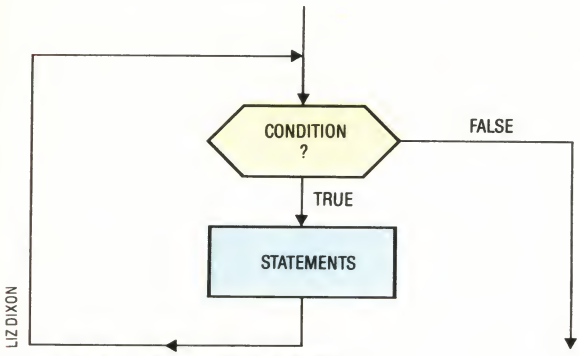
The other essential control structure (WHILE . . . DO) is not directly available in BASIC, but it can easily be simulated. WHILE . . . DO is a type of 'do-loop' and it means 'repeat a statement or set of statements WHILE a specified condition is true' or 'WHILE a condition is true DO something'.

WHILE . . . DO always tests the condition before the statements are executed, so if the test fails first time through, the statements (called the body of the loop) are not executed. As an example, consider a games program that prompts the player to 'PRESS SPACE-BAR WHEN READY'. This part of the program could be written (in 'pseudo-language' or simplified English) as:

```
WHILE space-bar is not pressed
DO scan keyboard
start game
```

In BASIC this could be written:

```
250 PRINT "PRESS SPACE-BAR WHEN READY"
260 IF INKEY$ < > "  " THEN GOTO 260
270 GOSUB *START*
```



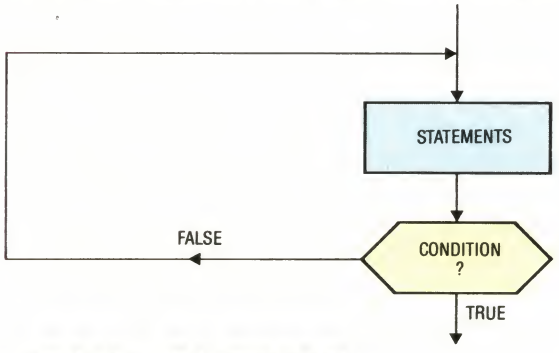
The DO . . . WHILE Control Structure
The loop is repeated as long as the condition is True. The statements may never be executed (if the initial condition is False)

Line 260 says IF INKEY$ is not equal to (< >) a space (" ") THEN go back and check the keyboard again. A slightly more elegant way of writing it would be:

```
250 PRINT "PRESS SPACE-BAR WHEN READY"
260 FOR X = 0 TO 1 STEP 0
270 IF INKEY$ = "  " THEN LET X = 2
280 NEXT X
290 GOSUB *START*
```

In this program fragment the loop (to scan the keyboard) is executed only if the space-bar has not been pressed. If the space-bar has been pressed (i.e. INKEY$ = " ") then the program exits from the FOR . . . NEXT loop to line 290, which is the call to the START subroutine. (NB. We are using 'labels' or names for subroutines. Many versions of BASIC cannot call subroutines by name and you will have to use line numbers instead of labels.)

We haven't encountered STEP before, and this is perhaps an unusual application for it. When using a FOR . . . NEXT loop, STEP allows the 'index' to be incremented in units other than one. FOR I = 1 TO 10 STEP 2 will cause I to have the value 1 on the first pass of the loop, followed by 3,5,7 and 9. The next increment (to 11) will exceed the limit of 10 so the loop will be finished. It is even possible to have the index counting backwards. For I = 10 TO 1 STEP–1 will cause I to count from 10 down to 1. Using STEP 0 is really a clever trick that ensures that the loop will never finish unless X is 'artificially increased' — as in the case of our IF . . . THEN statement.



The REPEAT . . . UNTIL Control Structure
The loop is repeated until the condition becomes True. The statements will always be executed at least once

Another useful control structure, again not directly available in BASIC but easily simulated, is REPEAT . . . UNTIL. Here the condition test comes after the main body of the loop, so the statement or statements in the main body will always be repeated at least once. Look at this 'random number generator':

```
10 PRINT "HIT THE SPACE-BAR"
20 FOR X = 0 TO 1 STEP 0
30 LET R = R + 1
40 IF R > 9 THEN LET R = 1
50 IF INKEY$ = "  " THEN LET X = 2
60 NEXT X
70 PRINT "THE VALUE OF R IS  ";R
```

Here, the main body (incrementing the value of R) is always executed at least once since the test to branch out of the loop (IF INKEY$ = " ") does not come until after the increment statement (LET R = R + 1).

Yet another non-essential but useful control structure is that usually called CASE. In BASIC, the CASE structure is implemented using either ON . . . GOTO or ON . . . GOSUB. This is how it works. ON . . . GOTO is a multiple-branching statement that incorporates several IF . . . THEN conditional tests into a single statement. Consider a program fragment that converts the numbers 1 to 7 into the words for the seven days of the week:

```
1050 IF D = 1 THEN GOTO 2020
1060 IF D = 2 THEN GOTO 2040
1070 IF D = 3 THEN GOTO 2060
1080 IF D = 4 THEN GOTO 2080
1090 IF D = 5 THEN GOTO 3000
2000 IF D = 6 THEN GOTO 3020
2010 IF D = 7 THEN GOTO 3040
2020 PRINT "MONDAY"
2030 GOTO*END*
2040 PRINT "TUESDAY"
2050 GOTO*END*
2060 PRINT "WEDNESDAY"
2070 GOTO*END*
2080 PRINT "THURSDAY"
2090 GOTO*END*
3000 PRINT "FRIDAY"
3010 GOTO*END*
3020 PRINT "SATURDAY"
3030 GOTO*END*
3040 PRINT "SUNDAY"
3050 GOTO*END*
```

A more compact way of achieving the same object in BASIC is to use ON . . . GOTO like this:

```
1050 ON D GOTO 2020,2040,2060,2080,
     3000,3020,3040
```

ON . . . GOSUB works the same way, except that the value of the variable determines which subroutine is branched to. Here is a slight modification of the dice program (see page 174) using ON . . . GOSUB to select the appropriate graphics for the dice selected by the RND function:

DECIMAL	BINARY	CHARACTER
32	0 0 1 0 0 0 0 0	=(space)
33	0 0 1 0 0 0 0 1	= !
34	0 0 1 0 0 0 1 0	= "
35	0 0 1 0 0 0 1 1	= #
36	0 0 1 0 0 1 0 0	= $
37	0 0 1 0 0 1 0 1	= %
38	0 0 1 0 0 1 1 0	= &
39	0 0 1 0 0 1 1 1	= '
40	0 0 1 0 1 0 0 0	= (
41	0 0 1 0 1 0 0 1	=)
42	0 0 1 0 1 0 1 0	= *
43	0 0 1 0 1 0 1 1	= +
44	0 0 1 0 1 1 0 0	= ,
45	0 0 1 0 1 1 0 1	= -
46	0 0 1 0 1 1 1 0	= .
47	0 0 1 0 1 1 1 1	= /
48	0 0 1 1 0 0 0 0	= 0
49	0 0 1 1 0 0 0 1	= 1
50	0 0 1 1 0 0 1 0	= 2
51	0 0 1 1 0 0 1 1	= 3
52	0 0 1 1 0 1 0 0	= 4
53	0 0 1 1 0 1 0 1	= 5
54	0 0 1 1 0 1 1 0	= 6
55	0 0 1 1 0 1 1 1	= 7
56	0 0 1 1 1 0 0 0	= 8
57	0 0 1 1 1 0 0 1	= 9
58	0 0 1 1 1 0 1 0	= :
59	0 0 1 1 1 0 1 1	= ;
60	0 0 1 1 1 1 0 0	= <
61	0 0 1 1 1 1 0 1	= =
62	0 0 1 1 1 1 1 0	= >
63	0 0 1 1 1 1 1 1	= ?
64	0 1 0 0 0 0 0 0	= @
65	0 1 0 0 0 0 0 1	= A
66	0 1 0 0 0 0 1 0	= B
67	0 1 0 0 0 0 1 1	= C
68	0 1 0 0 0 1 0 0	= D
69	0 1 0 0 0 1 0 1	= E
70	0 1 0 0 0 1 1 0	= F
71	0 1 0 0 0 1 1 1	= G
72	0 1 0 0 1 0 0 0	= H
73	0 1 0 0 1 0 0 1	= I
74	0 1 0 0 1 0 1 0	= J
75	0 1 0 0 1 0 1 1	= K
76	0 1 0 0 1 1 0 0	= L
77	0 1 0 0 1 1 0 1	= M
78	0 1 0 0 1 1 1 0	= N
79	0 1 0 0 1 1 1 1	= O
80	0 1 0 1 0 0 0 0	= P
81	0 1 0 1 0 0 0 1	= Q
82	0 1 0 1 0 0 1 0	= R
83	0 1 0 1 0 0 1 1	= S
84	0 1 0 1 0 1 0 0	= T
85	0 1 0 1 0 1 0 1	= U
86	0 1 0 1 0 1 1 0	= V
87	0 1 0 1 0 1 1 1	= W
88	0 1 0 1 1 0 0 0	= X
89	0 1 0 1 1 0 0 1	= Y
90	0 1 0 1 1 0 1 0	= Z
91	0 1 0 1 1 0 1 1	= [
92	0 1 0 1 1 1 0 0	= \
93	0 1 0 1 1 1 0 1	=]
94	0 1 0 1 1 1 1 0	= ^
95	0 1 0 1 1 1 1 1	= _
96	0 1 1 0 0 0 0 0	= `
97	0 1 1 0 0 0 0 1	= a
98	0 1 1 0 0 0 1 0	= b
99	0 1 1 0 0 0 1 1	= c
100	0 1 1 0 0 1 0 0	= d
101	0 1 1 0 0 1 0 1	= e
102	0 1 1 0 0 1 1 0	= f
103	0 1 1 0 0 1 1 1	= g
104	0 1 1 0 1 0 0 0	= h
105	0 1 1 0 1 0 0 1	= i
106	0 1 1 0 1 0 1 0	= j
107	0 1 1 0 1 0 1 1	= k
108	0 1 1 0 1 1 0 0	= l
109	0 1 1 0 1 1 0 1	= m
110	0 1 1 0 1 1 1 0	= n
111	0 1 1 0 1 1 1 1	= o
112	0 1 1 1 0 0 0 0	= p
113	0 1 1 1 0 0 0 1	= q
114	0 1 1 1 0 0 1 0	= r
115	0 1 1 1 0 0 1 1	= s
116	0 1 1 1 0 1 0 0	= t
117	0 1 1 1 0 1 0 1	= u
118	0 1 1 1 0 1 1 0	= v
119	0 1 1 1 0 1 1 1	= w
120	0 1 1 1 1 0 0 0	= x
121	0 1 1 1 1 0 0 1	= y
122	0 1 1 1 1 0 1 0	= z
123	0 1 1 1 1 0 1 1	= {
124	0 1 1 1 1 1 0 0	= |
125	0 1 1 1 1 1 0 1	= }
126	0 1 1 1 1 1 1 0	= ~

ASCII
Here is a complete list of the ASCII values between 32 and 126, their binary equivalents, and the characters they represent. The meaning attached to values outside this range varies considerably from machine to machine

```
390 REM SELECT SUBROUTINE
400 REM USING ON...GOSUB
410 ON D GOSUB 530,600,670,740,810,880
470 RETURN
```

Although your version of BASIC probably contains many statements and functions we have not covered, most will be extensions to the 'basic' BASIC designed to take advantage of particular features of your machine. Many of these will relate to graphics features built into the hardware — instructions such as PAINT, PAPER, INK, BEEP and CIRCLE. These tend to be 'machine specific' and so we have not included them in our course, though we will be giving you more details in other articles.

Before ending the basic part of our BASIC course, however, there are some loose ends to tie up — a discussion of the ASCII character set, together with a couple of functions for helping manipulate characters, and a way of defining new functions (or functions not included in your version of BASIC).

Several methods of representing letters of the alphabet and other characters such as numbers and punctuation marks in digital form have been devised over the years. One of the first was Morse code, which uses combinations of dots and dashes to represent characters. From the computer's point of view, Morse code suffers from the disadvantage of using different numbers of bits for different letters — between one and six dots and dashes for each character. Other attempts at making a more regular and systematic character code (e.g. the Baudot code, which uses five bits to represent up to 32 characters) have fallen by the wayside and the almost universal system now in use is the ASCII code (American Standard Code for Information Interchange).

The ASCII code uses one byte to represent the 94 printable characters, the 'space' and a number of control 'characters'. Eight bits could give 256 unique combinations (2^8), but this is far more than is needed to represent the characters of a standard typewriter or computer keyboard, so only seven are used, allowing for 128 unique combinations. (The eighth bit is usually wasted but is sometimes used to specify an alternative set of foreign language or graphics characters.) The binary and decimal ASCII codes for the standard range of characters are given in the table.

As you can see from the table, the ASCII code for the letter A is 65 and for B is 66. The codes for the lower case letters a and b are 97 and 98. Every lower case letter has an ASCII code value larger by 32 than its upper case equivalent. This constant 'offset' makes it easy to convert lower case letters in character strings into upper case letters, and vice-versa. To do this we will need two further functions not used so far in the Basic Programming course — ASC and CHR$.

The ASC function takes a printable character and returns its ASCII code equivalent, so PRINT ASC("A") would print the number 65 on the screen; PRINT ASC("b") would print 98.

The CHR$ function does the opposite; it takes a number, assumes it is an ASCII code and returns the character it represents. Thus PRINT CHR$(65) would print A, while PRINT CHR$(98) would print b. The CHR$ and ASC functions are widely used, along with LEFT$, RIGHT$ and MID$ in programs making heavy use of character strings. Here's a short program that accepts a character from the keyboard, checks to see if it is upper case and converts it to upper case if it is not:

```
10 REM LOWER TO UPPER CASE CONVERTER
20 PRINT "INPUT A CHARACTER"
30 INPUT C$
40 LET C = ASC(C$)
50 IF C > 90 THEN LET C = C -32
60 PRINT CHR$(C)
```

We shall see more of this type of string manipulation in forthcoming parts of the course.

Finally, in this round-up, a look at functions you may not have in your version of BASIC. Almost all versions of the language allow the programmer to create new functions, and these are almost as easy to use as built-in functions. The DEF statement signals to BASIC that a new function is being defined. Here's how to define a function to calculate the volume of a sphere (the formula is $V = \frac{4}{3}\pi r^3$, where r is the radius of the sphere and π (pi) is the constant approximately equal to 3.14159):

```
10 REM FUNCTION TO CALCULATE VOLUME OF A
   SPHERE
20 DEF FNV(X) = 4 * 3.14159 * X * X * X/3
30 PRINT "INPUT RADIUS OF SPHERE"
40 INPUT R
50 PRINT "THE VOLUME OF A SPHERE OF RADIUS
   ";R;" IS"
60 PRINT FNV(R)
70 END
```

This way of defining a function is fairly straightforward, but let's look at the line in detail:

```
DEFines  function identifier
   ↓     ↓
20 DEF FNV(X) = 4* 3.14159 * X * X * X/3
       ↑     ↑
FuNction  dummy variable
```

When the function is defined, the letters FN are followed by an identifying letter — V in the case of the function above — and this must then be followed by a 'dummy variable'. This dummy variable must also be used in the function definition on the right of the equals sign. When the function is used in a program, any numeric variable can be used in place of the dummy variable in the definition.

At a further point in the program above it would be equally possible to use the 'volume of a sphere' function like this:

```
 999 LET A = 66
1000 LET B = FNV(A)
1010 PRINT B
```

```
1020 LET C = 5
1030 LET D = B + FNV(C)
1040 PRINT D
1050 LET G = FNV(16)
1060 PRINT G
```

Some BASICS allow multiple variables to be used in the defined function. Thus, a function to find the average of two numbers could be written:

```
110 DEF FNA(B,C) = (B + C )/ 2
110 INPUT "ENTER TWO NUMBERS";B,C
120 LET A = FNA(B,C): REM THE 'AVERAGE'
    FUNCTION
130 PRINT "THE AVERAGE OF ";B;" AND ";C;" IS ";A
```

Notice that line 110 above combines the equivalent of two separate statements in one. Most BASICS will automatically print words appearing in double quotation marks following the INPUT statement, so this line is equivalent to:

```
110 PRINT "ENTER TWO NUMBERS"
115 INPUT A,B
```

Line 120 also manages to get the equivalent of two statements in one line by using the colon (:) separator. Statements that would normally belong on separate lines may be written on one line provided each 'stand-alone' statement is separated from the preceding one by a colon. This can help to save space in long programs, but its use is not to be encouraged as it makes programs less readable and mistakes more likely.

We have now covered all the main points of the BASIC language. In forthcoming parts of the Basic Programming course we shall look at program development and program design, rather than at the details of BASIC.

Answers To Exercises On Page 197

Bugs

An 'OUT OF DATA ERROR' will result, because there should be a total of 12 values in the DATA statement in line 130. Secondly, an error will arise in line 100 when it tries to address an element A(4,1). Line 100 should read:

```
100 PRINT A(X,Y)
```

Assigning Values

Below is one version of a program that will perform the required functions. Your own program may look different

```
10 DIM A(8,13)
20 FOR R=1 TO 7
30 FOR C=1 TO 12
40 READ A(R,C)
50 NEXT C
60 NEXT R
70 REM ADD TOTALS
80 GOSUB 300
90 REM PRINT REQUESTED DATA
100 GOSUB 200
110 PRINT "MORE DATA?"
120 PRINT "Y OR N"
130 INPUT A$
140 IF A$="N" THEN GOTO 160
150 GOTO 100
160 END
200 PRINT "WHICH MONTH?"
210 PRINT "1-FOR JANUARY,"
220 PRINT "13 FOR TOTAL, ETC"
230 INPUT M
240 PRINT "WHICH EXPENSE?"
250 PRINT "1-FOR PETROL"
260 PRINT "8-FOR TOTAL, ETC"
270 INPUT X
280 PRINT "VALUE IS ";A(X,M)
290 RETURN
300 FOR R=1 TO 7
310 LET T=0
320 FOR C=1 TO 12
330 LET T=T+A(R,C)
340 NEXT C
350 LET A(R,13)=T
360 NEXT R
370 FOR C=1 TO 13
380 LET T=0
390 FOR R=1 TO 7
400 LET T=T+A(R,C)
410 NEXT R
420 LET A(8,C)=T
430 NEXT C
440 RETURN
500 REM YOUR DATA FOLLOWS HERE
510 REM EIGHTY-FOUR VALUES
520 REM 'DATA 11.35, 9.87' ETC
```

Basic Flavours

ASCII

The Dragon-32 has a non-standard version of ASCII, which does not permit lower case characters, so the lower to upper case converter program will not convert lower case to upper case; try it anyway, and read your manual for more information about the Dragon-32 character set.

DEF FN(A,B)

On the Oric-1, the Vic-20, the Dragon-32 and the Commodore 64 you cannot write more than one variable inside the brackets.

ASC( )

On the Spectrum and the ZX81, replace:

```
ASC(A$) by CODE A$
```

and replace:

```
CHR$(65) by CHR$ 65
```

CHR$( )

If the argument is an expression, it must be put in brackets. Simple arguments — like A$ and 65 — do not need brackets, however.

ON... GOSUB

These statements are not available on the ZX81, Spectrum or Lynx.

ON... GOTO

INPUT" "

On the BBC Micro replace

```
INPUT "ANY MESSAGE";MS$
```

by

```
INPUT "ANY MESSAGE",MS$
```

DEF FN

On the BBC Micro you must define functions at the end of the program after the word END or STOP, not at the beginning as in the example given. In this case, as in many others, BBC BASIC is more powerful than standard BASIC, so consult your manual for more information.

STEP 0

On the Oric-1 in the two program fragments that demonstrate a good loop structure, replace:

```
IF INKEY$ = "  " THEN LET X = 2
```

by

```
IF KEY$ = "  " THEN LET X = 1
```

On the Dragon-32 replace it by:

```
IF INKEY$ = "  " THEN LET X = 1
```

On the Vic-20 and the Commodore 64 replace:

```
IF INKEY$ = "  " THEN LET X = 2
```

by

```
GET A$:IF A$ = "  " THEN LET X = 1
```

Dialling Tones

An acoustic coupler will convert digital data into audible tones and back again. Plugged into a telephone handset, it can be used to communicate with other computers

Connecting your home computer to a printer or a set of disk drives is relatively easy as they will usually be in the same room, if not on the same table. To connect your computer to its bigger brothers in your office or halfway round the world is a slightly different proposition. Fortunately, we already have a means of global communication — the telephone system. All that is needed is a means of connecting our computers to it.

Because the telephone system is so widely used by large computer systems (airline booking systems, for example), the technology is already well established. All the home computer user needs is a simpler and cheaper version of it. The conventional method of connecting computers to the telephone system is a device called a modem. This odd name is simply an acronym formed from the words modulator/demodulator.

The device works in much the same way as the cassette interface on a micro. The patterns of binary 1s and 0s are converted into electrical signals at two different audible frequencies and then sent down the telephone lines (this is the modulation process). At the other end they are 'demodulated' from audible frequencies back to 1s and 0s. The modem sends a constant frequency (called the 'carrier tone') whether or not data is actually being sent, which is how the receiving computer knows that the line is still connected.

The main disadvantage of a modem is that it has to be permanently wired into the telephone system, monopolising its use, which is something of an inconvenience for the home user.

An alternative method of communication is an acoustic coupler. Because the system uses tones that can be heard, there is nothing to stop these being generated acoustically, using a loudspeaker. This could then be coupled to the telephone handset and transmitted in that way. At the other end, a microphone placed in contact with the handset's earpiece could pick up the transmitted signal. This is what an acoustic coupler was designed to do. Unlike the modem, it does not need to be permanently attached to the telephone.

There are several types of acoustic coupler available, ranging from the device shown in the illustration, which is compact enough to be used with a portable computer, to large table-top units. Sophisticated units may be used to answer incoming calls automatically, without the need for a computer operator to be present, by constantly listening for the carrier tone. Just as cassette interfaces vary in the speed at which they can store and retrieve the information, so do acoustic couplers. The range of speeds is, however, strictly limited. The transmission characteristics of a telephone cable prevent any signal faster than 1,200 characters per second (cps) being transmitted with reasonable reliability.

Low-cost units may work at speeds as low as 30 cps, with more expensive models featuring a switch to select a variety of speeds. The important thing to remember, however, is that the devices at both ends of the telephone line must be operating at the same speed — otherwise no transmission can take place.

The tremendous increase in the use of personal computers in business and the removal of British Telecom's monopoly on the production of modems has resulted in the development of a large number of new products. Devices like the Sendata and its near relatives allow portable computers like Tandy's Model 100 and Epson's HX-20 to be used as remote computer terminals by anyone from sales representatives to journalists. Orders, articles and correspondence can be entered into the computer's memory and then sent down the telephone lines to head office.

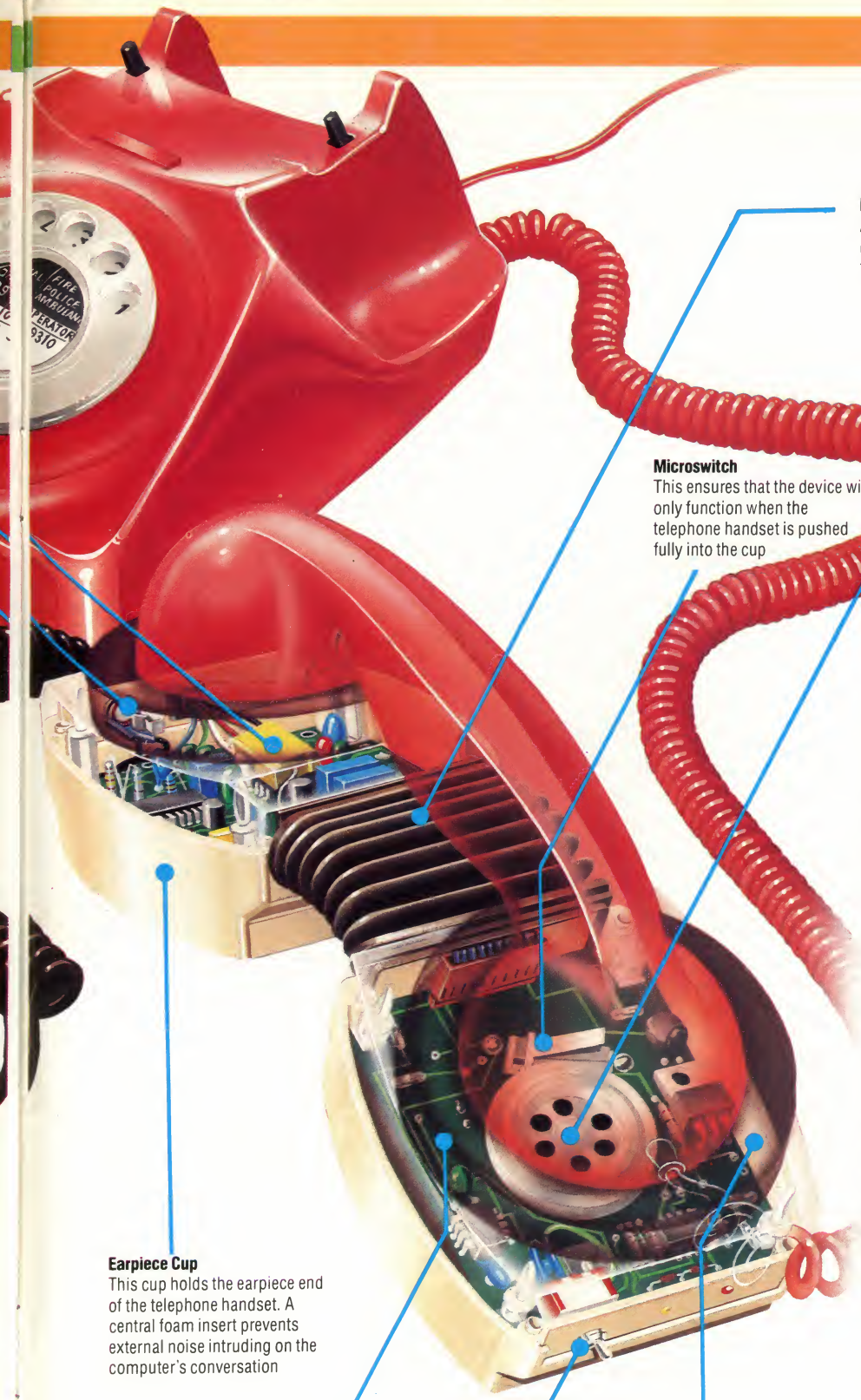
MARCUS WILSON-SMITH

Microphone
This picks up the signal from the telephone's speaker and feeds it to the circuit board

Power Socket
Supplies power to the coupler from a suitable transformer. It is also used to re-charge internal Ni-Cad batteries

Interface Cable
Connects to the computer's RS232 (serial) interface socket

Phone Boxes
Light-weight acoustic couplers allow the travelling computer user to be in communication with any other computer, anywhere in the world by means of the public telephone network



TREVOR HILL

In business a computer terminal and acoustic coupler allow instant access to a wide range of information services and computer bureaux. The ordering of supplies in one chain of chemist's shops is now completely computerised, with the staff entering stock items and quantities and then transmitting it to the main warehouse computer system. At home the acoustic coupler has the significant advantage over a conventional modem that it doesn't need to be permanently attached to the telephone lines. An executive working at home on a personal computer can contact his office to send or receive information without having the telephone line permanently tied up.

In the home computer market the acoustic coupler is providing a low-cost and convenient alternative to the conventional modem in allowing access to public databases such as Prestel and Micronet 800. They are also a much more reliable way of sending programs to friends than entrusting a cassette to the post. Electronic mail, using services like British Telecom's 'Telecom Gold', offers the home computer user an acoustic coupler with an instant communications facility of a kind that formerly only the largest companies had access to.

It is not possible, however, for a European acoustic coupler to speak to an American one. The American computing industry uses a system called Bell 103 and Europe uses a system called CCITT V21. Needless to say, they are incompatible.

A further complication is that the new generation of telephones currently being introduced don't seem to fit the cups (see illustration) on many acoustic couplers. Because the system works by transmitting sound, it is important to avoid the possibility of external noise getting into the telephone handset. If the telephone doesn't fit properly or there is a great deal of external noise, the data you send may very well be scrambled.

Common Knowledge

Computer 'workstations' located in the same building can be linked together using a Local Area Network. This enables several users to share common information and expensive peripherals

TONY SLEEP AT HARINGEY ITEC

Apple For Teacher
Networks are in common use at Information Technology Centres, like the one shown here, where young people train in the use of computers and microelectronics

Now that micros are becoming more and more common in schools and offices, the chances are increasing that there will be a number of compatible machines in the same building — or even in the same room. As soon as there are, someone inevitably turns his thoughts to methods of linking these machines, if only to share peripheral devices such as disk drives or letter-quality printers. Such add-ons as these are expensive and sharing them between a number of micros is an economical way of using them.

But the peripherals link-up is by no means the only benefit to be derived from joining machines together. Micros can be set up to communicate with one another in a 'network'. Where all the machines or 'workstations' are located within one building, the term 'Local Area Network' (LAN) applies.

Much has been said about the possibility of abandoning the use of paper in the office. An obvious first step is to replace the written memorandum with a message that appears from a remote source on a monitor screen. This 'postbox' facility is available on most Local Area Networks, and also on Prestel. Rather than disturb the recipient's work, the fact that a message is waiting is announced on the bottom line of the screen.

Another useful application is in the classroom. Text can be 'mirrored' from the teacher's micro to any or all of the students' workstations, or the teacher can use the facility to inspect any pupil's work, offering comments and suggestions.

The same facilities make possible the use of a common pool of information by a number of different users. Perhaps the most frequent application of this is to databases containing either public or private information — Prestel and the other viewdata systems are examples.

It is essential when a number of people are using the same 'file' of information to ensure that the information contained in it cannot be modified without all users being warned of the fact. For example, a manufacturing company might use a network of computers to hold information on the availability of components and raw materials. If each user is not presented with identical information, then at best there will be confusion, and at worst non-existent stocks will be allocated, perhaps to more than one department at a time.

Using microcomputers with perhaps 64 Kbytes of RAM or more, it is tempting to 'down-load', say, a portion of a database and then not refer to the master file again until you need to examine a different segment. Unless the controlling software is comprehensive enough to detect and reflect changes in information contained in that down-loaded section, the user may be referring to out-of-date figures.

The cost of much computer hardware is declining, but the technically more advanced peripherals can often cost more than the microcomputer that controls them. A good

example of this is the Winchester disk — a hard disk that runs in a hermetically sealed (airtight) environment, offering some of the speed and storage capacity of mainframe disk units. Winchester disks — named rather quirkily after the Winchester 30/30 repeating rifle, because they were originally constructed as twin disks, each with 30 megabytes capacity — are necessary for the volume of transactions handled by commercial software. They have enough spare capacity, however, even in very busy environments, to cope with more than one user accessing them at once. Similarly, while a dot matrix printer (see page 74) will be adequate for most jobs, word processing will often require a 'letter-quality' daisywheel printer costing considerably more. Unless one user can keep it constantly at work, it pays to be able to share such a printer across several workstations.

Networks offer other possibilities, too. Documents that require the attention of a number of people can be passed on from one to the other without being printed on paper. This happens, for example, in the more technically advanced newspaper and magazine offices. An author creates a typescript, which is then passed to the editor for his critical appraisal. Next it is the turn of the sub-editor, who corrects spelling, grammar and usage, marks the type specifications and passes it to the typesetter, who begins the printing process. Before Local Area Networks became available, this process meant that a typescript would often be retyped two or three times before it reached the printer.

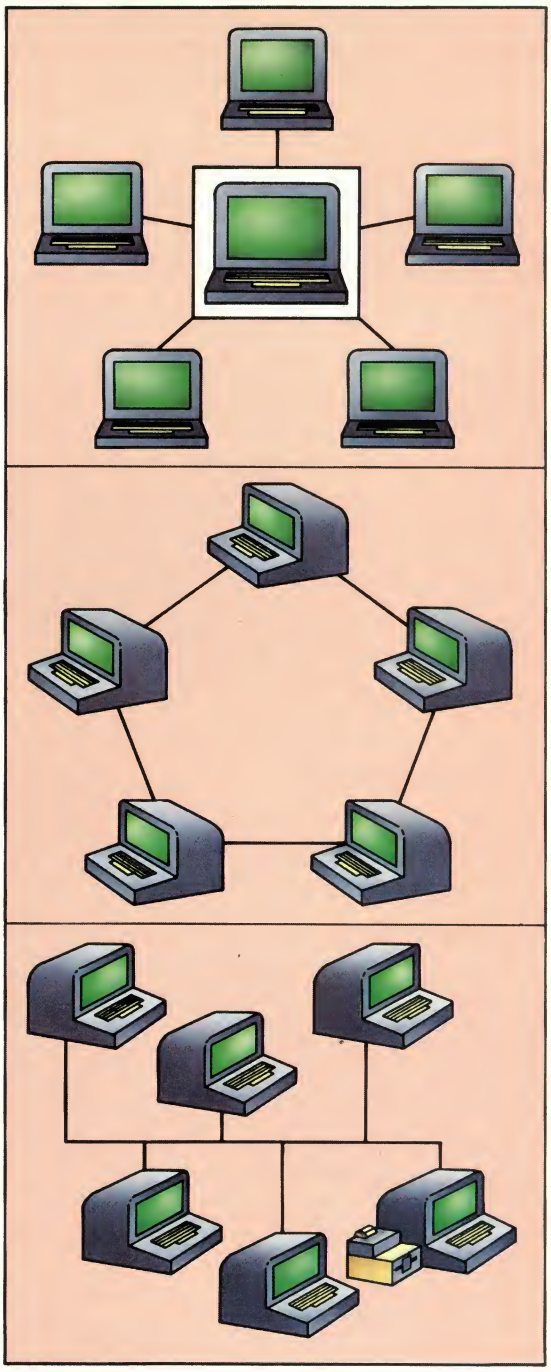
Network systems are available for most home micros, but perhaps the most widely used is Acorn's Econet for the BBC Micro. Econet designates one of the machines on the network to act as a 'file server', which looks after the central disk drive and all the various requests for information. This machine can be either dedicated to that purpose or available to a user whenever it is not needed to provide a service to other members of the network. If the network shares a printer there will need to be one machine available to control it.

Econet will support up to 254 workstations plus the two 'server' stations, but a much more realistic limitation on the size of the network is the distance that the farthest station can be away from the 'clock unit', a maximum of 500 metres (1,650 feet). The clock unit is a separate box that also sits on the network and controls the rate at which data is sent around the system. Econet uses two pairs of cables, just like the telephone system, and so is relatively simple to install. One of the pairs carries data, the other the clock pulses needed to ensure synchronisation.

Econet's rather sophisticated communications software resides in an eight Kbyte EPROM (Erasable Programmable Read-Only Memory) in every workstation. The system's most complex task is to prevent 'collisions', that is, to ensure that only one member of the network is transmitting at any one time. Other home computer-based networks are similar, though generally do not offer quite such comprehensive features.

Mini and mainframe computer-based networks have become widely used in the course of the last decade, and need not be restricted to one country. Many airlines, for example, use reservations and ticketing systems that span the entire globe, transmitting their data by telephone lines or satellite link.

With the general move towards cable television, it is reasonable to expect an increase in the use of networks, perhaps based on a similar concept to that of Micronet 800, a Prestel-based system that allows programs to be down-loaded over telephone lines onto the BBC Microcomputer.



KEVIN JONES

Types Of Network

Star
A star network connects each user's machine to a central controller, which will also manage the common peripherals

Ring
Some networks require the users' computers to be joined together in a continuous loop. They are less popular because the data might have to pass through as many as half the machines in the ring to get to its destination

Bus
The design of more refined networks, such as Econet, is very similar in concept to the architecture of a modern microcomputer: data and control messages are passed directly from one user to another

Charles Babbage

Though never fully implemented, Charles Babbage's machines were the forefathers of the modern computer

'I wish to God these calculations had been executed by steam!' Charles Babbage exclaimed as he laboured over the tables of the Nautical Almanac. The 19th century had developed steam power but accurate navigation at sea was still a problem. A ship's position was found by observing the moon and then using mathematical tables that were often inaccurate.

It was in 1812 that Babbage first had the idea of building a machine, which he called the Difference Engine, that could perform the laborious calculations needed for the nautical tables. By 1823 he had completed a small model and approached the government for a grant to build a working machine. The Chancellor of the Exchequer gave him £1,500 and he set about building an engine that would eliminate errors by automatically printing the results of its own computations.

Babbage's life's work was thereby determined. The project consumed vast sums of money, for he was working at the frontiers of the engineering skills available at that time. He obtained the money through the assistance of the Prime Minister, his friend the Duke of Wellington. Despite Babbage's confidence that 'whatever the engine did it would do truly', the government eventually withdrew, having sunk £17,000 into the venture. Babbage's engineer, Joseph Clement, resigned shortly afterwards, following an argument, and took with him all the tools that had been specially machined for the engine.

Babbage moved quickly onto a more ambitious project, the Analytical Engine, which was intended to accomplish all that the Difference Engine had been designed for and much more besides. Its design in many ways resembled that of the modern computer. It contained a memory store and an arithmetic 'mill' (equivalent to the CPU), provided printed output, and could even be programmed, using conditional branching.

At first the instructions were controlled by spikes as in a barrel organ; later the punched card system that Joseph Jacquard had introduced into the weaving industry was adopted. Babbage also experimented with different number bases but as all his machines were mechanical, there was no advantage to be gained from using the binary system.

He was joined in his project by his companion Countess Ada Lovelace, who was a gifted mathematician. They were dogged by problems, not least money. She lost much of her wealth gambling on an 'infallible' horse-racing betting system. After her death at the age of 36, Babbage continued alone.

A man of prodigious energy, he also invented the doctor's ophthalmoscope for looking into the eye, choreographed a ballet, devised a system of stage lighting and invented a technique for signalling at sea.

In his last years he grew irascible. Expecting a peerage, he turned down the baronetcy offered as recognition for his work.

Babbage's work anticipated the structure of the modern electronic computer but he failed to realise his vision fully. His Analytical Engine remained unfinished, its completion thwarted by the technical limitations of 19th-century engineering science.

1792
Born in Totnes, Devon, on 26 December

1810
Goes to Trinity College, Cambridge, to study mathematics

1814
Marries Georgina Whitemore

1822
Publishes a paper entitled 'Observations on the Applications of Machinery to the Computation of Mathematical Tables'. Receives the first Gold Medal of the Astronomical Society, which he helped to found

1827
Cambridge appoints him Lucasian Professor, the chair once held by Newton, at £80 a year, though he never lectures or takes up residence

1833
Parliamentary candidate at Finsbury

1834
Work on Difference Engine suspended after the engineer Joseph Clement resigns

1862
Partially complete Difference Engine exhibited at South Kensington, London

1871
Dies on 18 October

Top Of The Pops

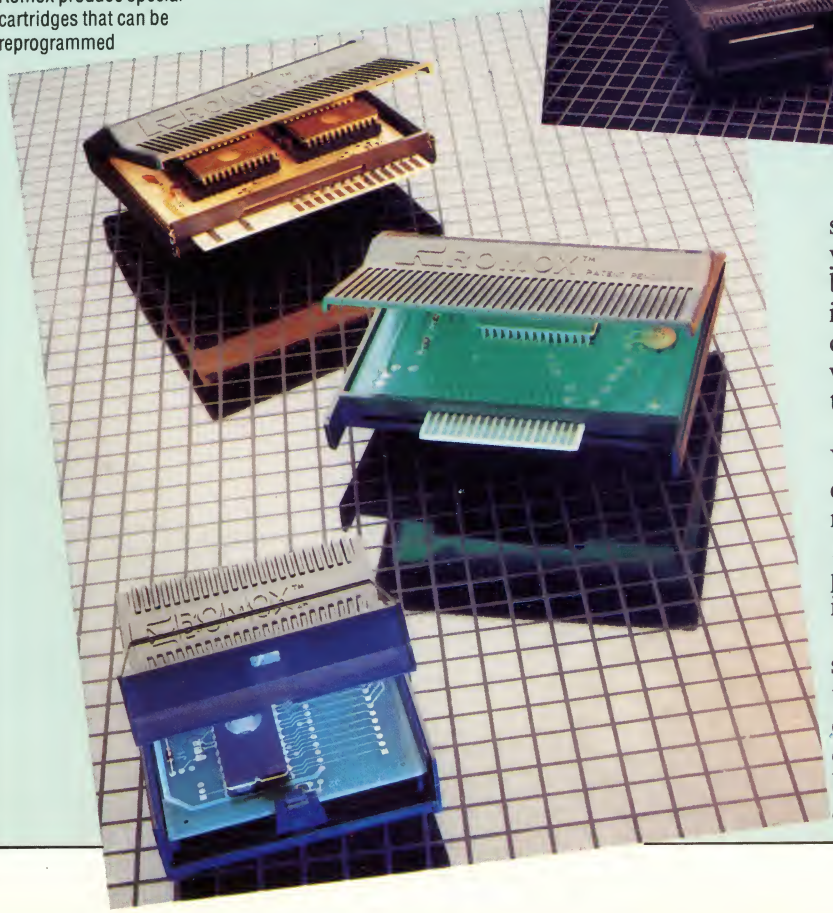
The problem with purchasing games software is that each time you change the game you have to buy a new cassette or cartridge. It would be much cheaper if people had to pay for the programs alone. One new way of doing this is a system called Romox, which could soon be appearing in shops selling software.

Romox is a new electronic distribution system that does away with the need for the shop to stock hundreds of different cassettes and cartridges and gives you all the software you want on one type of cartridge. The Romox machine is actually a microcomputer installed in the shop and plugged into the telephone network, by which it is linked to a mainframe computer. This can send hundreds of games over the telephone wire to the Romox micro, which stores them on a Winchester disk.

The customer takes a blank cartridge to the Romox micro and plugs it into one of its nine slots. Each slot is designed to match the type of computer the customer has at home. So if you have a Commodore computer you insert the blank Romox cartridge into the special Commodore slot; if you have an Atari the cartridge would go in the Atari slot, and so on. By pressing the control keys on the Romox computer a list of the current top 20 computer games appears on the screen.

The Romox games terminal is installed in the local computer store, and communicates with a central computer

Romox produce special cartridges that can be reprogrammed



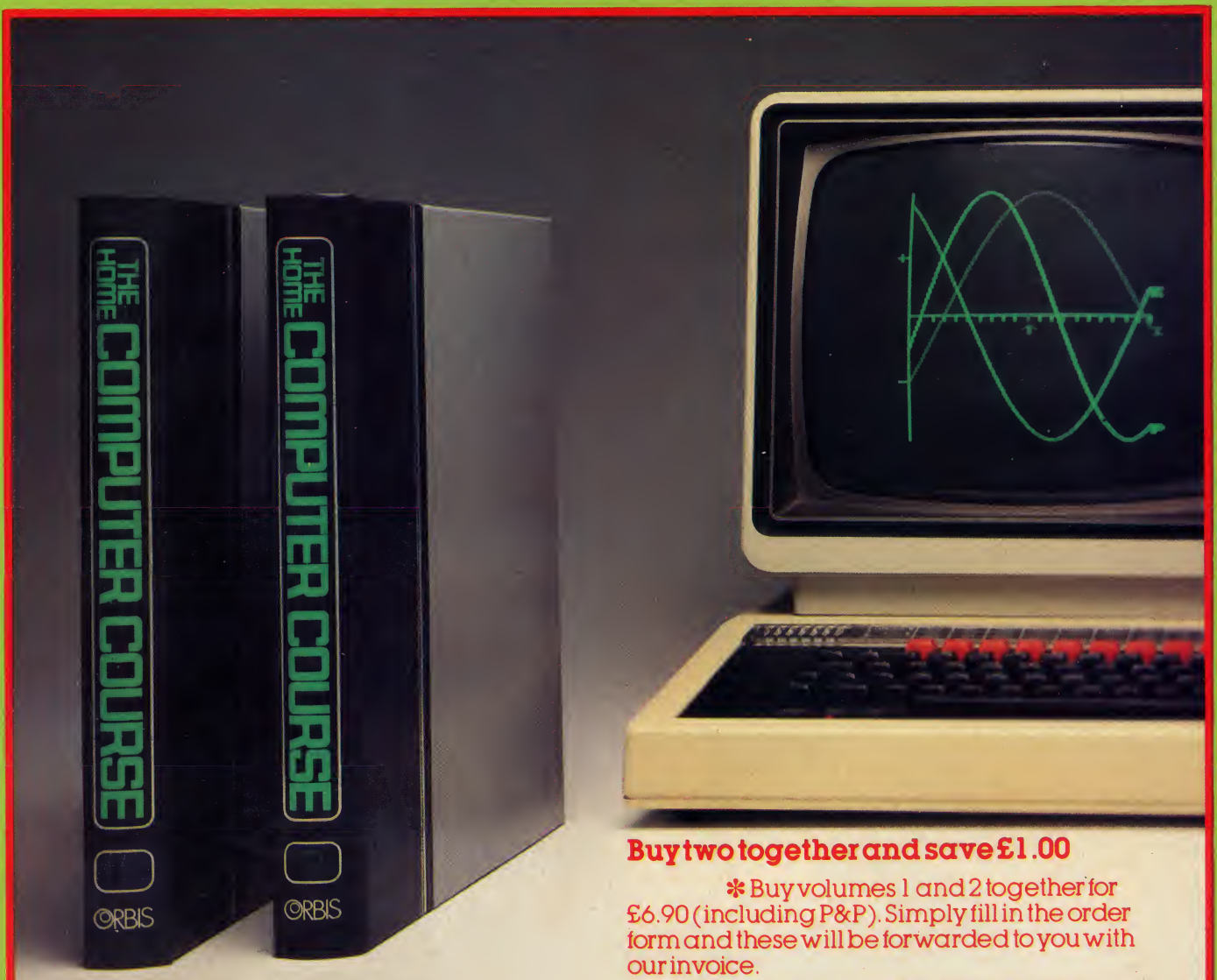
You choose your game and in less than a minute the cartridge is programmed. When you become bored with the game you simply take the cartridge back to the shop, plug it back into the Romox computer and choose another game. Of course, there is a charge for each game but there is no need to buy another cartridge.

The growing use of electronically distributed software could mean that loading a program into your computer is as easy as taping a radio broadcast. Instead of buying a program cassette it is now possible to receive software on your home computer by linking up to a central computer which then sends programs to your machine through the telephone system.

One such service, 'The Source', is based in Virginia in the USA. Subscribers can communicate with each other's computers and receive programs by using a modem (see page 108) on the telephone. They are given a special password, which gives them access to all the information stored.

In the UK the BBC is developing its Telesoft service. This sends programs to subscribers through a special adaptor linking the home computer and the television set. The programs are decoded in much the same way as Teletext broadcasts are decoded by television sets equipped with a Teletext adaptor.

THE HOME COMPUTER COURSE BINDER



Now that your collection of Home Computer Course is growing, it makes sound sense to take advantage of this opportunity to order the two specially designed Home Computer Course binders.

The binders have been commissioned to store all the issues in this 24 part series.

At the end of the course the two volume binder set will prove invaluable in converting your copies of this unique series into a permanent work of reference.

Buy two together and save £1.00

* Buy volumes 1 and 2 together for £6.90 (including P&P). Simply fill in the order form and these will be forwarded to you with our invoice.

* If you prefer to buy the binders separately please send us your cheque/postal order for £3.95 (including P&P). We will send you volume 1 only. Then you may order volume 2 in the same way – when it suits you!

Overseas readers: This binder offer applies to readers in the UK, Eire and Australia only. Readers in Australia should complete the special loose insert in Issue 1 and see additional binder information on the inside front cover. Readers in New Zealand and South Africa and some other countries can obtain their binders **now**. For details please see inside the front cover.

Binders may be subject to import duty and/or local tax.

THE LAST WORD IN LOGIC